COMPUTATIONAL TECHNIQUES FOR REASONING ABOUT AND SHAPING PLAYER EXPERIENCES IN INTERACTIVE NARRATIVES

A Thesis
Presented to
The Academic Faculty

by

David L. Roberts

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Interactive Computing

Georgia Institute of Technology
May 2010

COMPUTATIONAL TECHNIQUES FOR REASONING ABOUT AND SHAPING PLAYER EXPERIENCES IN INTERACTIVE NARRATIVES

Approved by:

Charles L. Isbell, Advisor
School of Interactive Computing
Georgia Institute of Technology

Mark O. Riedl
School of Interactive Computing
Georgia Institute of Technology

Andrea Thomaz
School of Interactive Computing
Georgia Institute of Technology

Brian Magerko
School of Literature Communication and Culture
Georgia Institute of Technology

Michael Mateas
Computer Science Department
University of California Santa Cruz

Date Approved: April 1st, 2010

ACKNOWLEDGEMENTS

Firstly, I would like to thank my parents. Your love and support over the years has been more valuable to me than you can possibly realize. For believing in me, for being my biggest fan, for your advice and guidance, thank you. Also my sister. For many of the same reasons I would like to thank you. Our conversations on "what it means to be a Roberts" have always given me perspective and kept me grounded. I love all three of you tremendously.

I would also like to thank my friends, those nearby and those far away. Those to whom I speak to regularly and the one to whom I may never speak again. In no small way, you guys have made my journey a brighter one. My heart-felt thanks go out to all of you.

To my advisor, Dr. Charles Isbell, I wish to express my gratitude for your support over the years. You gave me more than enough rope to hang myself with, but were always there to make sure my feet stayed on the ground. To the rest of my committee, Dr. Mark Ridel, Dr. Andrea Thomaz, Dr. Brian Magerko, and Dr. Michael Mateas, thank you for your insight and helpful input throughout the process. Your guidance has helped to shape my career in a profound way.

To the members of the PFUNK research group, you guys are awesome. From our lunch excursions to "the brain trust" and it's dominance at FICS, I'll always look back fondly at my time with all of you. Our many "strategy" sessions for scheduling snafus and insightful research conversations will also contribute to the lasting memories.

I would also like to thank Dr. Merrick Furst for the many insightful conversations and his significant input on the directions for my research.

To all of those whom I have not named explicitly here, my apologies. The list is long and space is limited. You all are also more than deserving of my thanks.

Lastly, part of this research was performed while on appointment as a U.S. Department of Homeland Security (DHS) Fellow under the DHS Scholarship and Fellowship Program, a program administered by the Oak Ridge Institute for Science and Education (ORISE) for DHS through an interagency agreement with the U.S Department of Energy (DOE). ORISE is managed by Oak Ridge Associated Universities under DOE contract number DE-AC05-06OR23100. All opinions expressed in this paper are the author's and do not necessarily reflect the policies and views of DHS, DOE, or ORISE.

PREFACE

This dissertation is organized into modular chapters. Where possible, forward and backward references between chapters have been minimized. The motivation for organizing this dissertation into self-contained chapters is to enable the reader to skip around as much as possible.

The introduction (Chapter 1) provides a summary of the entire dissertation, including an overview of interactive narrative, experience/drama maangement, and our contributions. As will be discussed in detail, we have decomposed the creation of an interactive narrative experience into five components: 1) story, 2) goal selection, 3) action/plan generation/selection, 4) action/plan refinement, and 5) the narrative environment. Each of these components of the problem is depicted in Figure 1—a roadmap for the reading of this dissertation.

The boxes at the top of Figure 1 indicate the part of the problem breakdown. Under each of the boxes are ovals that label the technical contribution of our work. Chapter labels are included in the figure to aid in guiding the reader. More specifically, here is how the document is organized:

I Introduction: This chapter explicates our ***thesis*** as well as highlights the ***research contributions*** of this dissertation. Background is provided including references to earlier work on the topic, a description of the ***Declarative Optimization-based Drama Management*** (DODM) formalism, and a detailed problem description and decomposition. The ***story*** level of the problem decomposition is addressed here

II Related Work: In this chapter, we will present a survey and qualitative analysis of contemporary systems for drama management. The goal of this chapter is to relate our approaches to the others in the literature, so the reader will have an idea of the relative strengths and weaknesses of the techniques we have developed

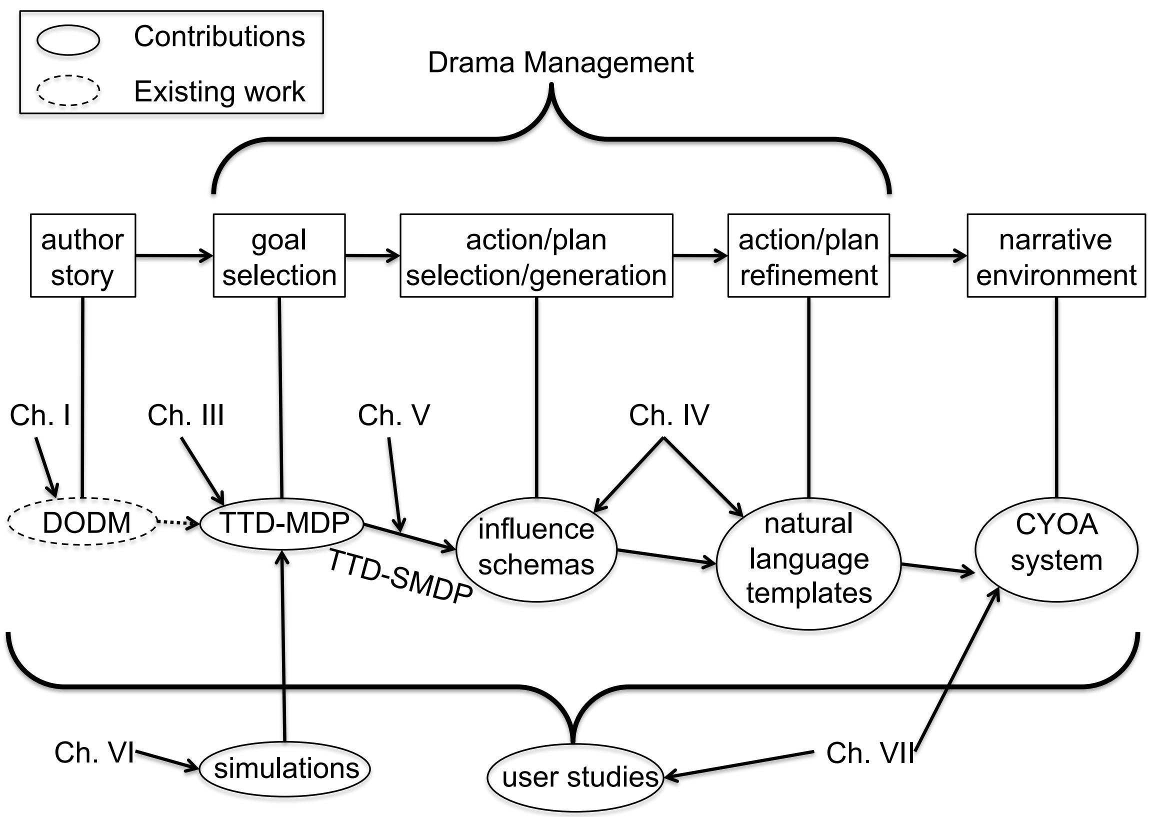


Figure 1: A depiction of the organization of this dissertation. This roadmap can serve as a guide for the reader wishing to skip certain chapters.

III TTD-MDPs: This chapter presents ***Targeted Trajectory Distribution Markov Decision Processes*** (TTD-MDPs), a formal framework for targeted non-determinism in online-decision making problems that was developed specifically to solve an instance of a DODM drama manager. TTD-MDPs address the ***goal selection*** level of the problem decomposition

IV Models of Influence: This chapter presents motivation for and formal ***models of influence and persuasion*** from social psychology as a technique for guiding player experiences in interactive narratives. It covers two of the five levels of the problem decomposition pertaining to ***action/plan selection/generation*** and ***action/plan refinement***

V TTD-SMDPs: This chapter presents an extension to the TTD-MDP model presented in Chapter 3 that illustrates that the algorithms used to solve a TTD-MDP will remain effective when ***non-atomic actions***, such as influence models, are used. It presents a model for ***Targeted Trajectory Distribution Semi-Markov Decision Processes*** (TTD-SMDPs) based on the theory of ***options*** from reinforcement learning. It addresses the gap between the goal selection and action/plan selection/generation levels of the problem decomposition

VI Simulation Results: This chapter is a focused discussion of the evaluation strategies used to characterize the performance of TTD-MDPs specifically. Here, we discuss the simulation environments we used to evaluate TTD-MDPs under theoretically "ideal" scenarios with computers rather than humans acting as players. The results of experiments on three different domains are presented

VII User Study Results: This chapter contains a detailed presentation of the architecture of our web-based choose-your-own-adventure-style interactive storytelling system. The findings from two studies using this environment are also presented and explained. This chapter is focused on the analyis and interpretation of the data from the studies. The majority of the un-interpreted data is included in various appendicies referenced throughout this chapter

VIII Conclusions and Future Directions: In this concluding chapter, we tie together each of the components discussed in the earlier chapters. We explicitly enumerate the research contributions of this dissertation as well as discuss future directions for continuing this research

Depending on the reader's interest in or prior knowledge of the various components of this document, certain chapters can be safely skipped without threat to understanding of the thesis. We encourage all readers to read the introdution (Chapter 1) and concluding chapter (Chapter 8). With the exception of Chapter 6 referring to the concepts in Chapter 3 and Chapter 7 referring to the concepts in Chapter 4, the remaining six chapters have as few references and dependencies as possible, so they can be read at will. For those readers interested in the technical aspects of this dissertation, we direct them to Chapter 3 and Chapter 5. For those interested in the social psychology aspects of this dissertation, we direct them to Chapter 4 and to some degree Chapter 7. The bulk of this document that is devoted to evaluation can be found in Chapter 6, Chapter 7 and Chapter 8.

The appendicies of this dissertation, of which there are six, are generally devoted to providing supplementary information. Appendix A contains a summary of our evluation of the related work surveyed in Chapter 2. Appendix B contains the details of our user study instruments. The complete set of results (without interpretation) for both user studies are in Appendix C and Appendix D. Lastly, we devote the final two appedicies to describing our authoring process the story we implemented for one of our user studies. Appendix E is a recounting of our efforts to implement an interactive story using our tools, looking back at the process as a case-study. Appendix F is a "data dump" of all of the inputs to our storytelling system. The reader interested in repeating our analyses should find everything necessary to implement our interactive story in this final appendix.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Interactive narrative experiences are marked by two important characteristics: 1) a large space of player interactions, some subset of which are specified as aesthetic goals for the system to realize; and 2) the affordance for the player to express self-agency and interact in a meaningful way with the system. As a function of those characteristics, players are (often unknowing) participants in the creation of the narrative experience. They cannot be assumed to be cooperative, nor can they be assumed to be adversarial. Thus, we must provide paradigms to designers that enable them to work with the players to co-create the experiences without the explicit transfer of the system's goals (as specified by the author) to players and without the need for the system to have a goal-oriented model of a player's behaviors or decisions. This dissertation formalizes compact representations and efficient algorithms that enable computer systems to represent, reason about, and shape player experiences in interactive narrative settings.

Early work in the field of interactive narrative relied heavily on so-called "script-and-trigger" systems, requiring sizable engineering efforts from designers to provide concrete instructions for when and how systems can modify or adapt an environment in service of providing a narrative experience to a player. While there have been advances in techniques for representing and reasoning about narrative experiences at an abstract level that automate (to a degree) the trigger side of script-and-trigger systems, few techniques have succeeded in providing designers with a paradigm that reduces their need for scripting system adaptations or reconfigurations—one of the contributions of this dissertation.

We first describe a decomposition of the design process for interactive narrative that induces three distinct technical problems: goal selection, action/plan selection/generation,

and action/plan refinement. This decomposition allows technical machinery to be developed for reasoning about the complete implementation of an interactive narrative. In turn, we describe representational and algorithmic solutions to each of these problems: a Markov Decision Process-based formalism for goal selection, a schema-based planning architecture based on theories of influence and persuasion from social psychology for action/plan selection/generation, and a natural language-based template system for action/plan refinement. To evaluate these techniques, we conduct experiments both in simulation and in an interactive storytelling system with human subjects.

In using these techniques, we realize the following three goals: 1) provide efficient algorithmic support for authoring interactive narratives for entertainment, education, or training; 2) design a paradigm for AI systems to reason about and act to shape player experiences based on author-specified aesthetic goals for interactive narratives; and 3) accomplish (1) and (2) with the player feeling more engaged and without the player perceiving a decrease in self-agency.

CHAPTER I

INTRODUCTION

Using concepts from narratology, interactive storytelling, and social psychology, we can design efficient algorithms and compact representations that enable computer systems to reason about narrative and shape human player experiences according to the aesthetic goals specified by authors. Software designers and authors have long struggled with balancing their artistic vision with the constraints or demands imposed by the hardware for which they develop. In recent years, the maturation of hardware has accelerated the increase in the demands that consumers put on the designers of computer-based entertainment experiences. In an effort to reduce the burden on designers, researchers have sought to develop paradigms that enable authors to create similarly complex experiences with less effort or increasingly complex experiences with the same effort [7, 74, 111].

In this dissertation, we will present our work in the areas of interactive storytelling and drama management [60]. Crawford [29] defines an interactive story as "a form of interactive entertainment in which the player plays the role of the protagonist in a dramatically rich environment." In other words, an interactive story is an experience in which a player interacts with an environment and the accumulation of their actions and changes in the environment bring about a strong notion of story. A drama manager (DM) is an agent for bringing about an interactive story. Loosely speaking, a drama manager is an omniscient coordinator that tracks player progress in a virtual experience. The coordinator proactively works to shape the player's experience according to the goals provided to it by the system's author. We will describe the ***Declarative Optimization-based Drama Management*** (DODM) formalism and use it to further motivate work using social psychology principles to inform the design of algorithms that automatically create and implement the actions available to the drama manager.

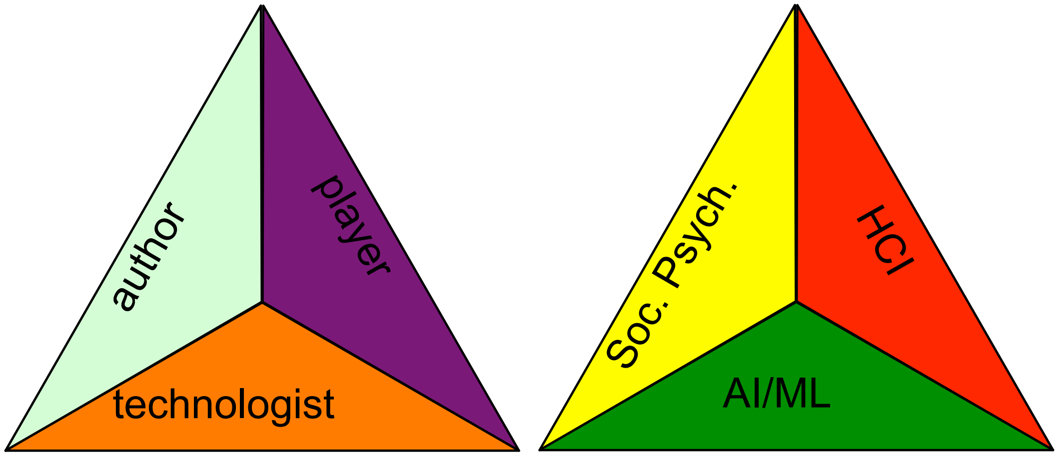
We will present a web-based choose-your-own-adventure-style system we used for evaluating our models and algorithms. The system has proven a useful tool for conducting loosely controlled evaluations for demonstrating the efficacy of our approach.

Succinctly, the thesis of this dissertation is the following:

Thesis: Using concepts from narratology, interactive storytelling, and social psychology, we can design efficient algorithms and compact representations that enable computer systems to reason about narrative and shape human player experiences according to the aesthetic goals specified by authors.

To test our thesis we have developed new concepts, algorithms, and models for drama management incorporating formal decisions processes, principles of influence and persuasion from social psychology, and representations from natural language generation to implement and evaluate an end-to-end automated drama management system. Our system is, to the best of our knowledge, the first to reason about not just what shape a player's experience should take, but how to go about realizing that shape without the player perceiving any change in their sense of self-agency. Our contributions can be grouped into three categories: theory, implementation, and evaluation. Briefly, the contributions of this dissertation are:

Theory: We present algorithms and representations that enable systems to reason about narrative structures in an interactive setting. The algorithms we have developed operate at various levels of detail in the specification of an interactive narrative, enabling systems to reason about abstract plot sequences as well as more detailed interventions in the narrative environment. The design of our algorithms was informed by techniques from statistical machine learning, social psychology, and natural language processing



(a) Three Constituents (b) Three Techniques

Figure 2: The problems we focus on are motivated by the requirement that we satisfy three types of users: authors, players, and technologists. The solutions we propose draw upon techniques in three domains: artificial intelligence and machine learning, social psychology, and human computer interaction.

Implementation: We have implemented algorithms, models, and a storytelling environment. They were implemented to test both in simulation and in practice. The storytelling environment provides a framework for examining how effective our algorithms are at realizing their purpose. It also provides a platform for conducting studies with human players

Evaluation: We provide an evaluation of each of our algorithms and our platform. We have run extensive simulation tests to characterize the performance of some of our algorithms along various dimensions of performance. Additionally, we use the narrative environment to test our algorithms in an interactive setting with human participants and report on these results

1.1 Three Constituents of Our Work

The work we describe in this dissertation is focused on the authoring process of interactive virtual experiences. That process has long been a cooperative effort among artists or designers and technical experts. More recently, as the increasing level of interactivity has become an important desideratum for these experiences, the player has been granted an increasing role in creating the ultimate experience. Therefore, the solutions we design must take into account the varying skills, desires, and needs of the three constituents: authors, technologists, and players (Figure 2(a)).

Authors have varying levels of technical expertise, but bring a creative knowledge to the design of the experiences. We need to develop authoring paradigms or tools that are sufficiently powerful to enable the creative expression of authors, but assume no particular level of technical competence.

In many cases, authors work with technologists to handle the implementation of their ideas. The job of the technologist is to write the code that will bring about the experience the author has envisioned. For technologists, our paradigms or tools should enable efficiency while retaining the power necessary to accomplish the creative goals of the authors. Since we cannot assume that authors will always have (or have access to) technical expertise, the algorithms we design must have simple controls that can be intuitively explained to non-technical experts, but also have more fine-grained controls that enable technologists to make more significant customizations when authors deem necessary.

Lastly, players are often unknowing participants in the creation of the experience. They can not be assumed to be cooperative, nor can they be assumed to be adversarial. Thus, we must provide paradigms to designers that enable them to build systems that work with the players to ***co-create*** the experiences without the explicit transfer of goals from authors to players.

To accomplish these tasks, we design paradigms and tools that draw upon concepts from three domains: artificial intelligence and machine learning, social psychology, and human computer interaction (Figure 2(b)). Each solution uses concepts from some or all of these three areas to meet varying needs in terms of technical sophistication or in terms of varying types of performance characteristics.

1.2 Three Design Problems

The difficulty of authoring computer-based interactive environments has received notable attention in the research community. As hardware technology continues to mature, possibilities for increasingly complex computer-based entertainment, training, or educational experiences arise. Along with those opportunities come new challenges to authors and designers. In this dissertation, we present a number of algorithms designed to support the authoring process for interactive experiences.

Commercial computer games produced today can be the result of tens, if not hundreds, of person years of a coordinated effort among artists, programmers, designers, and authors. An important goal in developing new technologies for authoring such games (and other similar interactive virtual experiences) is to provide a paradigm that preserves expressive power for authors without increasing either their authorial effort or their need for advanced technical expertise.

One approach to realizing this goal is to implement an omniscient coordinator that tracks the player's experience and adapts the environment to bring about a targeted progression of events. Such a *drama* or *experience manager* is tasked with guiding the player in a dramatic experience prescribed by the author [60, 106]. For the sake of consistency with the literature, we will refer to such managers as drama managers (DMs).

In the past two decades numerous approaches to drama management have been developed (see [74] for a survey of early work and [7, 111] for surveys of more recent work, as well as Chapter 2). Aside from implementing the environment and story itself, three design problems must be solved to fully implement a drama management system:

1. **Goal Selection**: The system must have a way of representing the state of the narrative and encoding (in terms of goals) the author's desired aesthetics for the experience. In addition, the system must have a way to reason about the player's behavior in the environment in order to select the appropriate narrative goals (see Chapter 3)

2. **Action/Plan Selection/Generation**: The system must have actions that provide it a way to affect the environment. More importantly, the system must be able to reason about how the actions it takes will affect both the player's experience and its ability to achieve the gaols the author specified for it (see Chapter 4)

3. **Action/Plan Refinement**: The system must have a way to ensure consistency of the actions it takes given the current state of the environment (see Section 4.3.2)

To date, the bulk of work on drama management has been focused on the goal selection problem. Various approaches have been designed and to varying degrees implemented and tested in simulation or with actual game environments. The approaches have used a range of representational schemes from bags of beats [73], to formal decision processes [96], to dynamic decision networks [90]. Similarly, the algorithmic reasoning solutions have run from AI planning [143], to optimization [135], to advanced statistical techniques [15].

Despite the significant representational and computational power provided by those approaches (and others), the systems have relied heavily on the author to implement solutions to the action/plan selection/generation and refinement problems. In this dissertation, we will discuss our results in designing an algorithm for drama management as well as our steps toward reducing the authoring complexity of the second and third design problems. Our approach is based on designing ***computational models of influence*** to allow drama management systems to reason about how to shape the player's experience and automatically create utterances that are both meaningful in the environment and will persuade the player to behave in a manner consistent with the goals the author has specified for the drama manager.

In subsequent chapters of this dissertation we will present an algorithmic model for drama management in the Declarative Optimization-based Drama Management (DODM) formalism [135, 95] called Targeted Trajectory Distribution Markov Decision Processes (TTD-MDPs) [112]. We will discuss extensions to that formalism that allow for non-atomic actions which model the use of influence. Additionally, we will present a set of computational models of influence some of which have been implemented and evaluated in a web-based choose-your-own-adventure-style interactive storytelling system.

1.3 Declarative Optimization-based Drama Management

The concept of a drama manager was first proposed by Laurel [60]. From a theater studies perspective, Laurel introduced the notion of a central coordinator that guided the actions or behaviors of actors to shape an interactive drama for an audience participant. In this formulation, the drama manager was a human director giving directions to human actors. Realizing a drama manager computationally has proven to be a complicated task, often requiring designers to make complex trade-offs between conflicting desiderata [111].

For our purposes, we will focus on one particular approach to a computational drama manager called the Declarative Optimization-based Drama Manager. First proposed by Weyhrauch [135] as a problem of pseudo-adversarial search, it has been reformulated and studied by a number of researchers in recent years as a formal decision making process [95, 96, 112, 108].

An instance of a DODM drama manager is characterized by four components:

- **Plot Events**: a set of important events that can occur during the narrative experience and precedence constraints that induce a partial order over event sequences that define the narrative

- **Player Model**: a probabilistic model of the progression of events that encodes the likely behavior of players in every story state

- **DM Actions**: a set of actions that the drama manager can take to guide the player down a desirable narrative path

- **Evaluation Function**: a mathematical encoding of what the author has specified as goals for the interactive experience, typically in the form of a linear function over features of event sequences

Given a particular virtual environment, the designer of the interactive experience can create a DODM drama manager by specifying each of those four components. Nelson *et al.* attained an important insight: the four components of a DODM drama manager correspond directly with the components of a Markov Decision Process (MDP) [96].

An MDP is defined by a set of states, a transition model, a set of actions, and a reward function. The partial or complete sequences of plot events in DODM are the states of an MDP, the player model is actually a probabilistic transition model, the DM actions are the actions, and the author's evaluation function operates as a reward function. As a result, reinforcement learning can be applied to DODM. The solution to an MDP is a policy that specifies the appropriate actions for the drama manager to take in very story state.

Citing issues of complexity and a desire for "predictably unpredictable" behavior by the drama manager, Roberts *et al.* [112] created Targeted Trajectory Distribution Markov Decision Processes (TTD-MDPs). TTD-MDPs provide a formal framework for efficiently authoring and solving a DODM instance, the details of which will be described in Chapter 3.

To date, the vast majority of the work on DODM has been abstract. Results have tended toward analysis of simulated environments with simulated players rather than complete working systems with actual human participants. There are a few exceptions [121, 122, 23, 97]. Sullivan *et al.* describe some of the difficulties they experienced when implementing a DODM drama manager in a playable game [121, 122]. For instance, the designer must manage the vast complexity of a continually changing story environment by carefully considering how the DM actions will affect the environment in numerous states. This is known as action "refinement" [135] and has been called a "Herculean" task. Despite its difficulty, it must be completed in order for a DODM instance to effectively manage a game.

A range of sophisticated artificial intelligence and machine learning algorithms have been developed for drama management [7, 74, 111]; however, all of these algorithms have focused on the goal selection problem and the various solution techniques for DODM are no exception. Solutions to the selection/generation and refinement problems have either been ignored when results have been stated in simulation or have been hand authored.

One of the arguments in favor of using a drama manager relies on the fact that the burden on authors to implement script-and-trigger systems has become unmanageable given the complexity of modern interactive experiences [18]. While the idealized notion of a drama manager can in theory reduce the burden on designers, current realizations essentially shift the complexity from the script-and-trigger logic to the specification of DM actions. That is, the burden on the author to specify an exhaustive set of DM actions for the intricate set of opportunities in which they can be applied is anecdotally a similarly-sized task to writing a script-and-trigger system.

Thus, in order for drama managers to become a realistic technology for authors to use, it will be necessary to devise algorithmic solutions to the selection/generation and refinement problems. In this dissertation, I will show that such solutions can be realized through the use of ***computational models of influence*** inspired by ideas from social psychology [26] and behavioral economics [6].

1.4 Summary of Dissertation

We develop AI algorithms for each of the three design problems of interactive virtual experiences by implementing and testing TTD-MDPs and computational models of influence for DODM. Each of the three levels of the drama management problem can be treated as a black box. DODM and TTD-MDPs (as well as earlier search and reinforcement learning methods) provide a solution to the goal selection problem. We propose a set of language-based models as solutions to the remaining drama management authoring problems. Ultimately, the goal is for the author to specify a DODM instance for a given virtual environment with some additional meta information and have the system automatically generate influence statements to shape the player's experience. For the purposes of this dissertation, we will focus on a proof of concept, taking the first few steps toward a highly autonomous system that can reason about and generate influence with little or no input from an author.

As will be discussed in Chapter 4, there are a large number of theories of influence in social psychology and behavioral economics. Our goal is not to develop a comprehensive theory of computational influence, but instead to create a simple model that will enable us to run controlled experiments to demonstrate the feasibility and effectiveness of our approach.

To generate influence, there are two steps that must be taken. First, the specific type of influence to be used must be identified. According to one of the many theories of influence, there are six basic principles: scarcity, liking, authority, reciprocity, social proof, and consistency [26]. Assuming this were the only theory of influence, a system using influence would have to choose the influence tool to use given the player and the situation in the environment. The specific choice of influence tool provides the system with an abstract plan for implementing it concretely. This process is a solution to the second drama management problem discussed above.

Given the choice of influence tool, the second step is to refine the abstract plan into a concrete set of utterances or world reconfigurations that will affect the player's decisions. This process takes as input the abstract plan selected or generated during the influence selection step and refines it given the system's knowledge of the player and the state of the environment or story. It is a solution to the third drama management problem discussed above.

A complete solution to both of these problems requires complexities that would more than likely obfuscate any results obtained during an evaluation. Therefore, we propose a simple template-based model to solve the refinement problem. Rather than design a complete solution to the selection/generation problem, we instead will hard-code a small set of solutions and appeal to the vast body of work in discourse planning (*cf.* [48, 85, 141]) as an indicator that a computational solution is feasible.

The remainder of this dissertation is organized as follows. Chapter 2 contains a thorough survey and qualitative analysis of related work from the literature. The qualitative analysis serves to highlight some of the design tradeoffs made in many of these systems and situates our approach in relation to others. In Chapter 3 we will make a detailed presentation of the TTD-MDP framework in the abstract, mostly divorced from the DODM problem it was designed to solve. The reader familiar with TTD-MDPs or uninterested in the technical details may wish to skip Chapter 3. We then will present a concrete set of models of the six types of influence [26] as solutions to the selection/generation problem in Chapter 4. In Chapter 5 we will describe extensions to the DODM and TTD-MDP framework that are required to ensure influence actions will not violate the conditions that enable the desirable theoretical properties presented in Chapter 3. In Chapter 6 we will present three simulation domains and the results of numerous simulation experiments that help to characterize the performance of the TTD-MDP formalism under "ideal" theoretical conditions. Then, in Chapter 7 we will present the architecture of the choose-your-own-adventure storytelling system we implemented to run our user studies. Further, we will analyze and report the results of two studies using that system: a study to verify the effectiveness of influence in a storytelling domain and an end-to-end evaluation of a fully implemented DODM drama manager. The latter confirms the result of the former that utilizing at least one method of social psychological influence enables a drama manager to affect the choices players' make in ways an author specifies as goals, without the players perceiving a decrease in their sense of self-agency. Lastly, Chapter 8 summarizes the contributions of this dissertation as well as includes a discussion of future directions.

CHAPTER II

RELATED WORK: DRAMA MANAGEMENT

In this chapter, we focus on the *interactive narrative* literature, a literature about entertainment experiences where a player is an active participant in how a story unfolds. An interactive story is an experience in which a player interacts with an environment and the accumulation of their actions and changes in the environment bring about a strong notion of story. A drama manager (DM) is an agent for bringing about an interactive story. Players exercise self-agency in their interaction with the stories by choosing to explore different parts of the environment, engaging other players or non-player characters in some way, and taking specific actions.[1] The environment (*e.g.*, objects in the world, the world itself, or other characters) reacts to the behavior of the player. This makes the experience interactive and player-driven. On the other hand, authors of these experiences design specific situations or plot sequences that they hope will occur during play. Thus there are goals specified by authors to create a *narrative* quality. It is the combination of these two features that creates interactive drama.

There is a natural tension between player self-agency and designer goals: realizing designer goals often necessitates removing player self-agency while ensuring player self-agency makes realization of designer goals difficult. In the earliest systems, authors addressed this tension using an exhaustive set of *local triggers* to provide instructions for the game world and non-player characters (NPCs); however, this approach simply does not scale [18]. Recently, the job of mediating this tension has fallen to a more centralized *drama manager* (DM), an omniscient coordinator that directs objects and characters in the

[1] Self-agency is a term with a definition that has undergone notable changes over the years [25, 33, 45, 75, 92, 133]. For our purposes, loosely following Wardrip-Fruin *et al.* [133] we will assume that a system that affords expression of self-agency is a system that makes available to the player the things they wish to accomplish.

game world to influence the plot progression. An omnipotent micromanaging drama manager that prevents any player actions corresponds to the traditional notion of drama while no drama manager corresponds to a fully autonomous experience. A DM that infrequently takes actions to influence—as opposed to modify deterministically—the experience corresponds to interactive drama. The ends of this spectrum are sometimes referred to as "strong-story" and "strong-autonomy" respectively [76].

Arguably the most famous example of a drama manager is that of the Façade drama manager [73, 77, 79, 80]. In Façade, the drama manager attempts to construct a narrative experience by creating dramatic tension. This is achieved by carefully selecting the set of plot events and the order in which they are presented in response to the player's interactions with the non-player characters (NPC).

The idea of using a manager to guide dramatic experiences was first proposed by Laurel [60]. Since then there have been a number of concrete implementations of the idea (see [74] for a somewhat dated survey and [111] for a more recent survey). In this chapter, we will survey a number of systems, focusing on more recent developments and discussing some of their similarities and differences. In addition, we provide a basis upon which to compare these systems—and more importantly upon which to situate the work presented in this dissertation. In particular, we describe a number of ***desiderata*** we feel are important metrics for the qualitative evaluation of these systems, and describe each system according to those metrics.

2.1 Desiderata

The subject of how best to evaluate a drama manager is a topic of some debate in the interactive drama community. A recent article by Rowe *et al.* proposes a framework for classifying types of evaluations for interactive narrative systems, but does not provide a comparative analysis of the existing approaches [117]. One concern arises from the need to separate the quality of result from the quality of the generative process. If it is found that players do not rate their experience more highly when a DM is used, it may just be that the author has created a deeply satisfying (or unsatisfying) experience and the DM cannot significantly change the quality of that experience. Alternatively, perhaps a drama manager could improve the experience if only the tools available to the author allowed her to be more expressive. Another problem arises when we try to separate the quality of the experience of authoring from the quality of the player's experience. It is not clear who has the highest priority. As we shall see, most systems assume just a model of player behavior and leave it at that.

In addition, there is a choice of perspective between system-building and analysis. Generally speaking, system builders are concerned with technical issues related to the process and problems associated with actual implementations of these systems. As such, some of the techniques surveyed in this paper are integrally tied to a particular game system. On the other hand, analysis is more concerned with looking at the features or affordances of a particular approach to drama management. These techniques tend to be presented independent of a particular game system.

For our purposes, we focus on analysis. Where it is possible, we have tried to separate the approach from the particular game system. Further, we assume that the author has created a generally pleasing narrative, so we can evaluate the drama management systems themselves. Note, however, that any analysis remains speculative in that our qualitative analysis characterizes the potential of a drama management system and the affordances it provides to open new avenues for authorship rather than characterizes the degree to which authors can actually exploit those affordances.

First and foremost, it is desirable for the drama manager to afford author's control as well as player's self-agency. These two qualities, however, are in service of a greater goal: to create a more engaging or believable entertainment/learning/training experience. In thinking about what, specifically, such a system should provide, there are a number of desiderata that come to mind. Beyond that, however, our specific choices for desiderata were motivated by three factors: 1) Our observations from building systems for managing interactive narratives; 2) The motivations discussed by the authors of the systems we survey (see [68] for example); and 3) Numerous discussions with researchers well versed in game and narrative rhetoric. They are:

- **Speed**: players should not perceive any delay in game action due to decision making by the drama manager
- **Coordination**: NPCs should coordinate to enhance the experience of the player characters
- **Replayability**: the game experience should be varied but retain high quality, even during repeated play
- **Authorial Control**: a DM should provide a way for an author to influence the experience of the player
- **Player Autonomy**: players should not be so constrained by the drama manager that they cannot pursue their own goals
- **Ease of Authoring**: the burden of authoring high quality dramatic experiences should not be increased because of the use of a drama manager
- **Adaptability**: a player's individual characteristics should be exploited to better the experience
- **Soundness**: the DM should be amenable to theoretical inquiry, allowing one to make verifiable claims about the system as a whole, not just about the underlying solution technique
- **Invisibility**: the drama manager should not appear overly manipulative to the player

- **Measurabilty**: the system should provide affordances for measuring author's satisfaction with the authoring process and the set of stories experienced by the player as well as the player's satisfaction

It is important to note that some of these desiderata are in conflict. For example, *player self-agency* and *authorial control* are known to be in tension [21, 70, 102, 109]. Riedl, Saretto, and Young [102], using the terms *control* (to mean player's control, rather than author's control) and *coherence*, point out that a player's control can threaten narrative coherence when the player's actions in the environment can affect the a drama manager's presentation of the story. When implementing a particular approach to drama management, a trade-off is unavoidable. Of course selecting an approach for any particular case is dependent on what is most appropriate for the particular application. Thus, in general, no one of the desiderata is more important than any other.

After describing the most pertinent systems published in the last decade, we will situate them in some detail with respect to two or three desiderata. In addition, we will describe the systems briefly for all 10 of the desiderata, classifying them into one of three categories: the system is well designed with respect to the particular desideratum (represented by ●); the designers did not engineer for this criterion (represented by ○); and the description of an approach in terms of a desideratum is highly implementation dependent (represented by ◐). We present a table summarizing each of the systems in Appendix A.

2.2 Drama Manager Components

To facilitate clearer comparisons, we briefly describe components common to all drama management techniques. All drama management approaches are based on: a representation of *plot*; a set of *drama manager actions* that can be taken in the game world; a *model of player responses* to DM actions; and a *model of the goals the author specifies*.

The representation of plot provides the basis upon which the drama manager can reason about the player's experience. Most representations are abstract, encoding only significant story events and possibly their relationships (*e.g.*, precedence constraints). In certain approaches to drama management, the representation of plot is comprised of operators that drama management can take in the environment. In other settings, plot is represented by NPC interactions with each other or the player, and their composition produces a story. Regardless of the specific choice, all drama management systems must have some way of representing plot.

Drama manager actions provide a way to steer a story toward a "good" sequence of plot points. These actions need not have direct concrete implementations in the game world. For example, a concrete DM action could be removing an object from the game world or causing an NPC to start a conversation. On the other hand, an action could be instructing an NPC to prevent a player character from crossing the street. In this case, the details of how to concretely accomplish this task in the game world are up to the (possibly semi-autonomous) NPCs.[2] Regardless of the implementation, the DM actions are the tools with which the drama manager influences narrative flow.

In order for the DM to reason about action selection, it must have a model of how actions affect the world. In particular, if the DM determines that a player is deviating too far from a desirable plot sequence, it must know which of the many actions available will best guide the player back toward a good sequence. Further, it must know enough to balance between gentle guidance that may not succeed and more heavy-handed actions that will succeed but may be overly apparent to players. For example, if the author wishes for the player to enter a particular building, the DM would not want to take an action to block the entrance, nor would it want to take an action that would clearly be herding the player into the building. Perhaps the DM would create an event that generates sounds from within the building, raising the player's interest in entering.

Finally, all DM systems must have a model of the goals the author specifies. The model

[2] As such, drama managers are similar to agent coordinators. NPCs are agents in a multiagent system communicating with a central coordinator to bring about a high level goal.

must be simple enough to describe and modify, but expressive enough so that the DM can choose proper actions.

2.3 Optimization-Based Systems

The techniques we describe in this section all use an optimization-based idiom for realizing author-specified goals. Specifically, these goals are specified in terms of an *evaluation function*. The drama manager selects from its available actions guided by the goal of optimizing this target function. Although originally rooted in traditional AI search techniques, current systems have borrowed heavily from statistical machine learning. This is in distinct contrast to the planning-based systems described later.

2.3.1 Search-Based Drama Management

Search-Based Drama Management (SBDM) is attributable to Bates [13] but was studied in greater detail by Weyhrauch [135]. SBDM is based on an abstraction of a game into significant plot events with precedence constraints encoded in a *directed acyclic graph* (DAG). The edges in the DAG do not imply that a particular plot point must occur immediately after its parent in the graph, only that if it occurs it must not occur before. Plot points are also annotated with information about the story such as the location in the story world where the plot point occurs or the dramatic tension that the player is likely to experience. Any sequence of plot points consistent with a topological ordering of the DAG is a valid story.

Game play in this framework proceeds in an alternating fashion with the player triggering plot events and the drama manager taking actions in response. The DM actions in this framework act on a particular plot point. The DM can: *cause*, *deny*, *temp_deny*, *reenable*, and *hint*. The cause action causes a plot point to occur in the game whereas a deny action prevents a plot point from ever occurring. The temp_deny action suspends a plot point from occurring until a reenable action is applied to it. The hint action should increase the likelihood that a particular plot point will occur. The DM can also choose not to act, allowing the player to be the sole influence on plot progression.

Player responses to DM actions are modeled as transitions between plot events. A coefficient is associated with each plot point. When a DM action hints at a certain plot point, the hint action has the effect of multiplying the coefficient associated with that plot point by a fixed amount. Then, the probability of the player experiencing a plot point is calculated by normalizing the coefficients associated with all of the plot points that have satisfied precedence constraints.

Lastly, the author supplies an evaluation function defined over a valid sequence of plot points and DM actions. In the literature, this evaluation function is defined as a linear combination of story features such as *activity flow*, *thought flow* or *manipulativity*. The output of this evaluation function is a measure of how good the story is in the eyes of the author—it does not reflect player preference.

Weyhrauch uses SAS+, a variant of the expecti-max game-tree search algorithm, to optimize the evaluation function. A tree structure is constructed by alternating levels of plot point nodes with DM action nodes. Search alternates maximizing nodes at the plot point levels with expectation nodes at the DM action levels. There are two variants. The first exploits symmetries in the story space to construct a memoization table that enables evaluations over complete stories to be propagated up from the leaves of the tree to interior nodes. The second is a fixed depth search that uses a set of sampled complete stories as a heuristic estimate of the value of the node at which the search terminates.

Lamstein & Mateas proposed revising this technique [59], and Nelson & Mateas further explored it by attempting to reproduce its results [93, 95]. More recently, Chen, Nelson, & Mateas [23] and Sullivan, Chen & Mateas [121, 122] have implemented a SBDM in a Zelda-like playable game. In this work, they uncover the difficulty that can arise when authoring a set of actions that will appear consistent with the situation in the game. For example, suppose one of the plot points occurs when an NPC starts a conversation. If the DM takes an action to cause that plot point when the particular NPC is not near the player, then the outcome could ruin the aesthetic of the story. To handle this situation, they add location tags as properties of actions. They were able to reproduce Weyhrauch's results, but found that the technique did not scale well.

Due to the combinatorial complexity of game tree search it is unsurprising that this system does not do well in terms of its **speed**; however, the designers took care to mediate this difficulty by imposing time limits on search and using heuristic evaluation. This system is especially **measurable**. Along with its derivatives described below, this approach to drama management provides a basis for characterizing the success of the drama manager in meeting the goals the author provides using the evaluation function. Evaluating this system typically includes calculating the frequency of the different function evaluations that are realized when the DM is used.

Evaluation of SBDM:

- **Speed**: ○, the combinatorial complexity of full-depth game tree search is intractable
- **Coordination**: ◐, this is an abstract system and coordination is implementation dependent
- **Replayability**: ○, the only non-determinism arises from random sampling (for the heuristic evaluation) and is not principled or controlled
- **Authorial Control**: ◐, affordance provided by causers and deniers gives high degree of control, but is implementation dependent
- **Player Self-Agency**: ◐, if sufficient hints are authored for DM actions, the player can exercise self-agency
- **Ease of Authoring**: ○, authoring in the abstract narrative domain seems appropriate, but describing quality in terms of linear evaluation over features is untested as of yet
- **Adaptability**: ○, does not model or adapt model of player to inform DM decision making

- **Soundness**: ○, nature of sampling for static evaluation does not provide affordance for theoretical investigation
- **Invisibility**: ◐, the concrete implementation of the abstract DM actions will determine invisibility
- **Measurability**: ●, the author's evaluation function provides a solid basis to characterize performance

2.3.2 Declarative Optimization-Based Drama

Nelson *et al.* continue work on SBDM by introducing *Declarative Optimization-based Drama Management* (DODM) [95, 96]. In this work, the plot point abstraction, DM actions, player transition model, and author evaluation function are exactly as in SBDM; however, the SAS+ sampling search is replaced with a policy obtained by solving a Markov Decision Process (MDP). MDPs provide a mathematical framework for modeling an online decision making problem when the dynamics of the world are stochastic [53]. An MDP is specified by a set of states, actions, a stochastic transition model encoding dynamics, and a reward function. The solution to an MDP is a policy dictating the choice of action in every state that will maximize the long-term expected reward. In this formulation of a drama manager, each of the components corresponds to a piece of an MDP specification. The current history of plot points and DM actions define state; the DM actions define a set of actions; the player model defines a probabilistic transition model; and the author's evaluation function defines a reward function. The solution to the MDP represents the optimal choice of action for the DM given any history of plot points and DM actions.

Unfortunately, reinforcement learning is susceptible to local optima, a phenomenon common to optimization techniques. Due to the stochastic nature of the game dynamics, it is likely that the computed policy will not be optimal. Thus, *Self-adversarial Self-cooperative Exploration* (SASCE) was developed to help find solutions to MDPs that best avoid "bad" parts of the story space. The idea behind SASCE is to use the current estimate of the state-value function that defines the MDP policy to select player transitions that are adversarial. In other words, the actual player model is not used in learning the SASCE policy. Instead a "self-adversarial" player model is substituted that forces the DM to learn a policy that optimizes for the worst possible player behavior. Results obtained by simulating game play against the actual player model indicate that this approach helps to reduce the frequency of poorly rated stories while increasing the number of moderately rated stories.

In contrast to SBDM, DODM has an advantage in terms of runtime **speed** because a policy specifying drama manager actions for every situation is learned before game play; however, it does require significant offline computational effort. Like SBDM, it also provides an affordance for **measurability**. Further, reinforcement learning is theoretically well-grounded and **sound**. Experiments suggest that DODM improves performance; however this appears to come at the cost of **replayability**. The system finds a narrow set of good stores and drives the player towards them.

Evaluation of RL for DODM:

- **Speed**: ●, RL-trained policy means action selection is simply a lookup, rather than a computation; however, offline computation can be quite expensive
- **Coordination**: ◐, like SBDM, DODM is abstract and coordination will be author dependent
- **Replayability**: ○, deterministic optimization limits variety of experience
- **Authorial Control**: ◐, if authors take advantage of cause and deny DM actions
- **Player Self-Agency**: ◐, with the use of the hint DM action the author can provide for increased player self-agency
- **Ease of Authoring**: ○, authoring abstract narratives seems feasible, but it is unclear if authors think in terms of linear combinations of story features

- **Adaptability**: ○, one player model is used to describe all player types and it is not adapted during game play
- **Soundness**: ●, the MDP formalism provides theoretical underpinnings
- **Invisibility**: ◐, subject to quality of concrete implementation of author specified DM actions
- **Measurability**: ●, the author's linear evaluation function provides a solid basis to characterize performance

2.3.3 TTD-MDPs

In this section we discuss the merits of some of the technical contributions of this dissertation. *Targeted Trajectory Distribution MDPs* (TTD-MDPs) are a variant of MDPs developed specifically to address the issue of replayability [15, 21, 22, 108, 109, 112].[3] A TTD-MDP is defined similarly to an MDP by: a set of trajectories that represent sequences of MDP states; a set of actions; a stochastic transition model; and a target distribution specifying a desired probability for every complete trajectory. The solution to a TTD-MDP is a *stochastic* policy providing a *distribution* over actions in every state such that under repeated play the sequence of states will match the target distribution as closely as possible.[4] Any finite-length discrete-time MDP can be converted to a TTD-MDP by simply encoding the history of MDP states into the TTD-MDP trajectories. This results in a TTD-MDP where each trajectory represents a sequence of states in the underlying MDP, optionally including a history of the actions taken.

The specification of authorial goals is a bit trickier in TTD-MDPs. Thus far, there have been two approaches taken: converting the DODM-style evaluation function and using

[3]The work of van Lent *et al.* also seeks to address replayability using a two level planning system: a strategic or deliberative level and a tactical or reactive level [132]. Unfortunately, this approach is designed for adversarial games and seems ill-suited to plot-driven open world games where drama managers are typically used.

[4]Closeness is typically determined by an error measure such as L_1 norm or KL-divergence.

a set of prototype trajectories. The techniques discussed here are presented in detail in Section 3.4.1 and Section 3.4.2.

Evaluation-based: Roberts *et al.* present a method for converting the author's evaluation function into a probability distribution over stories [112]. Because the evaluation function is not typically generative, they present an approach that estimates a target distribution. First, a set of stories is sampled uniformly—ignoring stories that evaluate too poorly—and used to construct a "trajectory tree." Probability mass is assigned by normalizing the evaluation scores across all the leaves in the sampled tree. These probabilities are then propagated up the tree to produce a probability for partial stories. Thus, when the DM selects actions according the probabilistic policy that is solved for, it is actually targeting stories in proportion to their evaluation quality.

Prototype-based: Roberts *et al.* extend TTD-MDPs by introducing an alternative authorial idiom based on a pre-specified set of desirable stories [21, 108, 109]. In this work, they replace the conversion process with a mixture of Gaussians (MOG) model. Rather than define a function that attaches value to a story, the author specifies a set of good prototype stories and defines a distance measure between stories. Each prototype becomes the centroid of a (possibly multivariate) Gaussian distribution. The probability mass that represents the "desirability" of a story is assigned by first determining its distance from each centroid.

This approach is amenable to even more **authorial control**. Specifically, each prototype can be treated differently, assigning unique (potentially non-uniform) mass in the MOG and unique variance along distinct dimensions. Thus, the authorial question becomes that of providing a small set of desirable stories and indicating a level of desirability. Further, the extent of the Gaussian can be tweaked to emphasize different aspects of stories. In this model, the author can adjust the allowed deviation in any direction by adjusting the values in the covariance matrix associated with each centroid. On the other hand, the work presented in this dissertation on the use of social psychology influence for generating

DODM actions seems to be beneficial for player-self agency.

TTD-MDPs have proven quite good at addressing **replayability**. Unfortunately, there is potentially a cost in the **ease of authoring**. Defining distributions by inferring them from an evaluation function is no more difficult—but also no easier—than defining an evaluation function in other DODM approaches. Providing prototypes may be easier; however, it is unclear if authors will find it easy to define game-specific distance measures that capture the nuances of their intent.

Evaluation of TTD-MDPs for DODM:

- **Speed**: ●, can be solved online with a convex optimization technique
- **Coordination**: ◐, as with SBDM and DODM, coordination is dependent on the implementation
- **Replayability**: ●, targeted non-determinism gives the author control over variety of experience
- **Authorial Control**: ◐, subject to the use of cause and deny DM actions
- **Player Self-Agency**: ●, subject to the use of the DM hint action
- **Ease of Authoring**: ●, the prototype-distance authoring idiom provides an intuitive paradigm for specifying authorial goals
- **Adaptability**: ○, the universal player model does not adapt to different players to change DM decisions
- **Soundness**: ●, a greedy online solution has been proven optimal
- **Invisibility**: ◐, subject to concrete implementation of abstract DM actions
- **Measurability**: ●, in the sampling paradigm, the measurements from SBDM and DODM are inherited; in the prototype-distance paradigm, divergence from the target distribution can be calculated

2.4 *Planning-Based Architectures*

Optimization-based approaches are predominantly derived from statistical machine learning methods. In this section, we discuss other approaches that have roots in AI planning techniques.

2.4.1 Narrative Mediation

Narrative mediation is a technique where a story is defined by a *linear plot* progression and by player choices. These components induce a story structure that is modeled as a partially ordered plan. The basic idea is to pre-compute every way the player can violate the plan and generate a contingency plan. The collection of all contingency plans and the narrative plan form the *narrative mediation tree*. To prevent unbounded mediation trees, certain player actions are surreptitiously replaced with "failures." This is similar to the "boundary violations" discussed by Magerko in the context of IDA (see Section 2.5.1 below).

The initial narrative plan represents the author's ideal story. In this sense, narrative mediation is similar to prototype based TTD-MDPs. It can be proven that this method of authoring interactive narrative is equally as powerful as creating branching story graphs. Here we discuss two systems that implement narrative mediation: Mimesis and the Automated Story Director.

2.4.1.1 Mimesis

Young *et al.* have developed the Mimesis system [24, 102, 139, 140, 142, 143], a planning system for drama management. A fairly complex architecture, Mimesis is primarily a run-time behavior generator. Mimesis works at multiple levels of abstraction and brings together both the procedural representations used by game engines and the declarative representations used by AI planning systems. In contrast to the architectures described earlier, Mimesis does not select the goals to pursue; it develops plans that are implemented at various levels of abstraction in the game to achieve the goals that are selected for it.

In contrast to some of the other approaches such as IDA (see Section 2.5.1 below) which are proactive, Mimesis is reactive. Suppose the player obtains an object that an NPC needs in order to carry out a plan. If the NPC continues with its existing plan, it will fail. To account for this, Mimesis will either repair the NPC's plan through re-planning or alter the effects of the player's actions to prevent it from obtaining the object. Note that Mimesis will not predict that a player will take an action to threaten a plan; however, it will notice that the outcome of an action taken in the world threatens an existing plan.

As mentioned above, Mimesis constructs plans at multiple levels of abstraction. In a functioning system, the request for a plan comes from the game engine, in the form of a set of goals and actions in the story world. The request is handled by the story world planner. This level is implemented using DPOCL, a hierarchical refinement planner. The story plan is then passed back to the game engine and to a discourse planner [141]. The game engine executes the parts of the story plan that pertain to characters, objects in the world, and the environment in general. The discourse planner constructs a complementary plan to control the music, camera angles, and other auxiliary aspects of the game experience. The combination of the story plan and the discourse plan form a coherent narrative plan that, when executed by the execution manager, will achieve the game engine's requested goals.

Mimesis is similar in nature to IDA; however, it allows more **player self-agency**. On the other hand, it lacks **invisibility**. The failure mode of this approach can easily result in an intervention that is apparent to the player.

Evaluation of Mimesis:

- **Speed**: ◐, as with IDA, re-planning is expensive in any sizable domain; however, Mimesis can pre-compute contingency plans if configured to do so, which contributes significantly to speed improvements

- **Coordination**: ●, the combination of procedural and declarative representation planners enables for a coordinated top to bottom experience

- **Replayability**: ○, like most systems, is reliant on the player as the only source of non-determinism

- **Authorial Control**: ●, the dual planner approach provides an affordance for high authorial control

- **Player Self-Agency**: ●, the reactive, rather than proactive, nature of the planning systems allows higher degrees of self-agency

- **Ease of Authoring**: ○, obtaining consistency from two unrelated planners can require significant authorial effort

- **Adaptability**: ○, does not model or adapt to the player's specific behaviors

- **Soundness**: ○, lacking in provable qualities

- **Invisibility**: ○, the combination of the story and discourse plans can make for an obvious intervention by the DM

- **Measurability**: ○, there is no affordance for measurability in this system

2.4.1.2 Automated Story Director

Riedl *et al.* implement narrative mediation for a cultural training simulation [102, 103, 104, 105, 107, 143]. This believable agent architecture, known as the *Automated Story Director* (ASD), has two goals: first, it must provide instruction to autonomous believable characters that help to shape the player's experience in the neighborhood around the narrative training goals; and second, it must monitor the story world to detect any inconsistencies that arise as a result of player actions and repair the narrative plan accordingly. To accomplish this, they modify the "failure" semantics discussed above to change the narrative goals of the system rather than simply fail.

This system shares a lot in common with IDA and Mimesis. If you consider the spectrum from reactive to proactive enclosed by Mimesis on one end and IDA on the other, then ASD lives somewhere in the middle. ASD also shares some similarities with the beat-based drama manager of Mateas & Stern (see Section 6.2); however, in contrast to beat-based systems where non-determinism and loosely specified authorial goals provide distinct player self-agency appropriate for narrative situations, this system uses a planning based approach to "recover" authorial goals when player actions change the narrative flow. The ASD approach is well suited to training or learning environments where player self-agency is intended to support exploratory learning rather than improve the quality of the entertainment experience.

ASD is **theoretically sound**. To our knowledge, this is the only system for which theoretical properties explicitly pertaining to narrative rhetoric (as opposed to mathematical properties of the solution) have been proven. Additionally, the handling of **player self-agency** is laudable, because contingencies for achieving authorial goals are *modus operandi*. On the other hand, the only source of **replayability** comes from player choices.

In addition to ASD, the Mimesis system also performs narrative mediation. Whereas ASD uses a completely pre-specified narrative mediation approach where all contingency plans are computed in advance, Mimesis accomplishes this through a complicated caching and speculative re-planning scheme. The Mimesis approach requires that all characters in the game (including the human player) obtain permission from the mediator before executing their actions. Thus, rather than fully determining all contingency plans, Mimesis can cache those most relevant to the current narrative plan and construct new ones as player actions move the narrative toward parts of the story space not as heavily represented by mediation plans in the cache.

Evaluation of ASD:

- **Speed**: ●, pre-computation of the narrative mediation tree results in solid online performance (except in catastrophic cases where re-planning is required); however, offline computation can be significant
- **Coordination**: ◐, dependent on the implementation

- **Replayability**: ○, the player is the sole source of non-determinism
- **Authorial Control**: ●, the mediation tree enables the system to ensure the author's goals are met
- **Player Self-Agency**: ●, the use of the mediation tree enables the system to react when players threaten the narrative path, giving a sense of self-agency
- **Ease of Authoring**: ○, authoring in STRIPS-like planning domain requires competence in AI techniques
- **Adaptability**: ○, does not model or adapt to the player's specific behaviors
- **Soundness**: ●, things have been proven about the representational power of the mediation tree
- **Invisibility**: ◐, is dependent on the "repairs" the author provides
- **Measurability**: ○, no affordance is provided for measurability

2.4.2 Dilemma-based Narratives

Barber and Kudenko [11, 12] have developed a system based on the notion that "drama is conflict". It dynamically generates *dilemma-based* interactive narratives. The narratives are potentially infinite in length and adapt to both the evolving relationships between the characters and to the player's behavior. To induce dramatic tension in the narratives, the player is coerced into making decisions based on clichéd dilemmas found in typical modern soap operas. These dilemmas are woven together using an overarching story line.

The system has three components: a *knowledge base* for characters, actions, dilemmas, and the environment; a *model of the player's* behavior and preferences; and a *narrative planning system*. The narrative environment is defined by the knowledge base. Specifically, the characters themselves are defined by attributes such as relationships with other characters and principles such as kindness toward others. Through the use of dilemmas, the characters are often forced to choose between conflicting principles such as greed and loyalty. In addition, the actions available to characters are specified in advance as part of the knowledge base.

The authors have identified five basic dilemma types that can ground out in any number of domain-specific ways: *betrayal*, *sacrifice*, *greater good*, *take down*, and *favor*. In the betrayal case, the character must choose whether to take an action that increases their utility while decreasing the utility of a friend. The converse is the sacrifice case. Similarly, there is the greater good case where the character has to choose whether to take an action that will increase an enemy's utility as well as their own. The converse is the take down case. In the favor case, the character must choose whether to take an action that is personally neutral, but increases the utility of another character. Each domain-specific dilemma is annotated with preconditions as well as utility changes for the characters involved. The specific characters involved may be determined at runtime.

To construct the narrative, the system selects among the set of available dilemmas based on an appropriateness estimate as well as the frequency with which each particular type of dilemma has been employed already. Appropriateness is determined mostly by the ongoing modeling of the player's behavior under specific dilemma types. Using this model, the system estimates how the player would act in a particular dilemma and then estimates how difficult the dilemma will be for the player. The dilemma that will be most difficult is most likely to be selected. Given a selection, a story world planner constructs a plan to satisfy the preconditions of the dilemma. Upon realization of those preconditions, the player is forced to decide the outcome of the dilemma. The system then reacts and selects the next dilemma to present to the player. In addition to player dilemmas, the system can create "character dilemmas" between two NPCs. These are used to help set the stage for the player dilemmas.

This system is notable for its **adaptability** in modeling the player and using that model to select among dilemmas. Additionally, the use of the planner to bring about dilemmas in a manner that forces **coordination** of NPCs is laudable. Unfortunately, this power brings about an increased **authorial burden**.

Evaluation of dilemma-based narratives:

- **Speed**: ○, online planning approach can be slow in any sizable domain
- **Coordination**: ●, is designed to coordinate NPCs to bring about dilemmas for player characters
- **Replayability**: ●, the system dynamically creates narratives both independently of the player as well as in response to their actions
- **Authorial Control**: ●, domain engineering allows for high degree of authorial control over dilemma types, frequencies, and applicability
- **Player Self-Agency**: ●, player is free to avoid dilemmas and act in whatever manner they feel like
- **Ease of Authoring**: ○, requires STRIPS-like specification of the domain and character specific information, which necessitates AI competence
- **Adaptability**: ●, models players and chooses dilemmas for them by trying to maximize their expected utility
- **Soundness**: ○, there is no affordance for theoretical inquiry
- **Invisibility**: ●, although this system forces players to make decisions, the "soap opera" genera for which is it designed obfuscates the work of the narrative generator
- **Measurability**: ●, can measure influence of the DM on the modeled player utility value

2.5 *Non-Planning and Non-Optimization Systems*

In this section, we evaluate a number of approaches to drama management that are neither optimization-based nor planning-based. The technical approaches underlying these systems vary greatly, ranging from probabilistic graphical models to case-based reasoning.

2.5.1 Interactive Drama Architecture

Magerko & Laird present the *Interactive Drama Architecture* (IDA) [67, 68, 69, 70, 71]. In their system, narrative goals are defined by the author at varying degrees of detail and the job of the drama manager (called the story director) is to ensure that the player's actions do not threaten their realization. For example, suppose the author specifies a goal for a particular NPC to provide an object to the player near the end of the story. If the player meets this particular NPC early in the game and chooses to fire a gun at it, the story director must intervene to prevent the bullet from killing the NPC. IDA uses semi-autonomous SOAR agents [56] that enable the directions from the DM to be made at various levels. Thus, in this case, the DM could instruct an agent to simply "prevent the death" of the NPC and allow the agent to determine how. On the other hand, the DM could provide specific instructions such as "make the pistol jam." In either case, a successful outcome preserves the author's goals.

In this system, plot events are labeled with preconditions in the form of logical statements. This approach supports dynamic runtime binding. For example, plot events can be authored with a variable, x, that appears throughout the story. When the player causes the first plot event using x to occur, it is bound to a concrete entity in the game world. This ensures that all subsequent plot events using that variable preserve narrative consistency while minimizing *authorial effort*. This type of runtime adaption is not a feature of the optimization-based systems described above.

Additionally, these logical statements can indicate temporal extent: particular plot events can have a range of discrete times between which they must occur. Thus, if a player is too early or too late in causing a plot event, the DM will recognize this as a threat to preconditions and can intervene. Interestingly, there is no notion of *explicit causality* in IDA. In other words, the DM cannot cause plot events to occur, but can prevent player actions that will preclude plot events from occurring. IDA reasons about *potential threats* using a *predictive player model*. Thus, the game world is a large unstructured space. But, through proactive modification of the game world, the drama manager limits the player to the portion that is consistent with the author's specified narrative goals: the player has complete self-agency provided they remain within the scope of the narrative goals.

IDA's most significant quality is **invisibility**. One side effect of IDA's approach is a potential increase in the player's perception of self-agency. This characteristic is subjective and has not been explicitly measured. Similarly, some aspects of **ease of authoring** also remain unmeasured. It is an open question whether the non-expert can easily construct predictive player models.[5]

Evaluation of IDA:

- **Speed**: ○, use of a planner that is reliant on online re-planning can be slow
- **Coordination**: ●, the use of semi-autonomous SOAR agents provides an affordance for good coordination if authored properly
- **Replayability**: ○, the use of a deterministic planner will bring about the same narrative structure repeatedly
- **Authorial Control**: ○, the lack of causality in this system makes authorial control very difficult
- **Player Self-Agency**: ●, use of DM as a mediator allows for sandbox like exploration of the game environment by the player

[5] In the work described here, the author constructs the model by hand. Mott, Lee & Lester have worked on predicting player goals by learning probabilistic models [88].

- **Ease of Authoring**: ●and ○, the requirement of an accurate predictive player model can be very difficult to author whereas the use of runtime variable bindings can reduce the specification burden on the author

- **Adaptability**: ○, does not consider player's goals when making action choices, only tries to ensure the narrative is consistent with authorial intent

- **Soundness**: ○, nothing has been proven about this system

- **Invisibility**: ●, designed to be proactive and lightweight so the player does not perceive any influence by the DM

- **Measurability**: ●, small evaluation of the DM's influence

2.5.2 U-Director

Mott & Lester [90] developed *U-Director*, a narrative planning infrastructure that is designed to deal with the uncertainty in narrative environments induced by player self-agency. Their goal is to develop a system that satisfies what they call *narrative rationality*, defined as reasoning in a principled manner about narrative objectives, story world state, and user state in the face of uncertainty to maximize narrative utility.

The "director agent" ensures plot progress and narrative consistency using *dynamic decision networks* (DDNs). DDNs are a generalization of Bayesian networks that include utility and choice nodes as well as time-varying attributes. The network is constructed using a level of abstraction similar to that of SBDM, DODM, and TTD-MDPs where DM actions are abstract directions that can have any number of concrete implementations in the game world.

Mott & Lester define a narrative decision cycle that is characterized by three levels of a dynamic decision network: the current game state (characterized by a decision node); the game state after the director's action has been taken (characterized by a chance node); and the game state after the player's reaction (characterized by a utility node). The utility nodes represent authorial goals in much the same way that the evaluation function does for SBDM and DODM. Each of these levels of the network contains nodes that represent details about the game and the players. The decision network contains nodes for the player's goals and beliefs (or knowledge gained about the salient facts of the story through interaction) as well as experiential state (or degree the player has been manipulated by the DM and how engaged they are in driving the plot). To actually make a decision, the director updates the narrative state according to the structure of the network in each of the three time slices associated with the current decision cycle. With the network updated, the director can perform action selection by analyzing each action's influence on the utility node in the third time slice.

In their tests, Mott & Lester have a network with 200 chance nodes, 400 causal links, and 7,000 conditional probabilities as well as a separate network of 50 nodes to express narrative utility preferences. It seems unlikely that the non-expert will find this **easy to author**; however, this approach is theoretically well-grounded in the body of work on dynamic decision networks and so is quite **sound**.

Evaluation of U-Director:

- **Speed**: ○, inference in Bayesian networks can be slow (albeit more efficient that modeling the complete joint)
- **Coordination**: ◐, is dependent on the concrete implementation
- **Replayability**: ○, non-determinism is modeled in the system, but not leveraged in DM decision making to target a variety of experiences
- **Authorial Control**: ◐, it is dependent on the style of actions the author provides
- **Player Self-Agency**: ◐, it is dependent on the style of actions the author provides
- **Ease of Authoring**: ○, dynamic decision networks are a fairly advanced machine learning technique and require specific knowledge of probabilistic graphical models

- **Adaptability**: ●, the explicit modeling and adaption of player relationships, experience, and utility influence decision making

- **Soundness**: ●, relies on the theories behind probabilistic graphical models

- **Invisibility**: ◐, is dependent on the specific implementation

- **Measurability**: ●, the use of utility nodes in the decision networks enables claims to be made about performance

2.5.3 Beat-Based DM

Mateas & Stern define a narrative to be a sequence of events that induce "changes in values." These values are properties of individuals or relationships such as love, hope, or anger. They define a *beat* as the "smallest unit of value change" and a *scene* as a "large-scale story event" [78]. Computationally, a scene in an interactive narrative is defined by a number of annotations: a set of preconditions; the values that are changed during the scene; a large collection of beats to effect the desired change in values; and a temporal description of how the values should be changed during the scene. Thus, an interactive narrative is defined by a set of scene definitions.

With scenes as the basic building blocks, Mateas & Stern develop a *beat-based drama manager* and implement it in their interactive fiction Façade [73, 77, 79, 80]. The drama manager is provided with a desired *global plot* arc that defines a shape for the change of the dramatic variables. The DM first determines the set of scene definitions that have satisfied preconditions and selects the one that matches the current position of the global plot arc as closely as possible. Then, the DM maintains a bag of beats associated with the current scene and reactively applies them until it realizes the desired value changes for the scene. Note that the change on dramatic values by a particular beat is a function of the beat's characteristics and the human player's participation. Thus, beats define an expectation over value change.

This authorial idiom is unique among all of the drama management systems surveyed in this paper. Due to the level of granularity required to author beats and their interactions, a beat-based drama manager seems ideally suited to the small-world variety of dramas like Façade; however, the freedom of **replayability** and **authorial control** may come at the price of **ease of authoring**, at least for large systems.

Evaluation of Beat-based DM:

- **Speed**: ●, the simple search through bags of beats is all that is required for DM decision making
- **Coordination**: ●, is specifically designed so beats affect coordination between the two NPCs in the narrative environment
- **Replayability**: ●, random selection of beats that meet the current requirements enables variety of experience (although it is not as controllable as one might hope)
- **Authorial Control**: ●, the highly detailed specification of value change associated with beats and scenes enables a high degree of authorial control
- **Player Self-Agency**: ●, the player-interaction determines the value changes so it will further affect the DM's choice of appropriate beats
- **Ease of Authoring**: ○, the level of detail required of annotations can present a significant authorial burden
- **Adaptability**: ○, no model of the player is maintained during episodes; this system relies on the author's description of player behavior
- **Soundness**: ○, there is no affordance for theoretical inquiry of this system
- **Invisibility**: ◐, this is highly author dependent
- **Measurability**: ○, the effect of the beat-based DM cannot be quantified

2.5.4 OPIATE

Fairclough implements a narrative story generation system called OPIATE. OPIATE uses a *story director* to drive narrative events in an open environment where the story is generated in real-time in response to the changing game environment and the player's actions [36]. The story director has a "world view" about the state of the game, using that to construct plans to achieve dramatic goals. It uses a *case-based planner* that is endowed with a plan library created using expert knowledge of skeletal plot structures and how they fit into the story world.

The case-based planner uses its dramatic goals and plan library to synthesize plot-based and character-based stories. A k-nearest neighbor algorithm is used for case retrieval that additionally provides a "suitability" score for each of the retrieved cases—the most suitable case is the sub-plot that should be enacted given the current state of the story world and the current state of the characters (including their attitudes toward each other and the player). A "suitability threshold" is used to determine if the best case should be used or cases should be combined to create a new case to be enacted by the story director. The suitability score can be decomposed to provide an individual score for each "function" in the case. Thus, case combination is simply a matter of finding the highest scored set of functions and combining them to form a new case. Once a case is selected, a "casting" approach is used where the abstract instructions of the case are assigned to specific characters based on defined roles. For example, if the role of "hero" is embodied by the player, then the NPC that opposes the player the most will be cast as the "villain." Thus, as the relationships between the characters change throughout the dramatic experience, the cases that are retrieved change based on the suitability of the casting of the characters pertaining to their relationships. This is similar to the work of Mateas & Stern on beat-based drama management where the scenes that are selected by the DM are chosen based on their fit to the dramatic values that represent the characters and their relationships.

There is notable **authorial effort** required to construct a case-base for the OPIATE system. On the other hand, its unique approach to dynamically casting non-player characters into different roles based on evolving relationships encourages **replayability** and provides a unique form of **coordination**.

Evaluation of OPIATE:

- **Speed**: ○, the choice of representation and size of case library can cause the planning system to perform slowly
- **Coordination**: ●, narrative decisions specify roles for each NPC to play in the environment
- **Replayability**: ○, as the case-library evolves, the choices made by the DM will first become more varied and then become more static once a sizable enough case-library has been developed
- **Authorial Control**: ●, the casting approach taken gives a high degree of control to the author allowing for specific narrative events to be forced to occur
- **Player Self-Agency**: ●, the player controls their relationships with other NPCs, which influences the evolution of the game
- **Ease of Authoring**: ○, requires notable effort to annotate sub-plots with relationship information as well as to develop a large enough case library
- **Adaptability**: ●, the choices of the system are made based on the player's evolving relationship with the NPCs
- **Soundness**: ○, the system provides no affordance for theoretical inquiry
- **Invisibility**: ◐, this is dependent on the specific set of sub-plots authored, but seems that the task of authoring for invisibility can be accomplished easily
- **Measurability**: ○, no affordance for measurability is provided by this system

2.5.5 Player Preferences

Sharma *et al.* have taken an approach to drama management that explicitly includes a model of player preference in the DM's decision making [119]. Drawing a distinction between *player preference* models and *player action* models, they identify one criticism of many other methods: drama management techniques overwhelmingly use artificial models of player behavior that do not explicitly represent the player's preferences or goals.

This approach is based on a simplification of the SAS+ algorithm that nonetheless extends it by combining the author's evaluation of a story and the player's preference for that story. They employ a *case-based reasoning* (CBR) system to determine player preferences by comparing their behavior to the behavior of earlier players. Preferences are elicited through a series of evaluation questions after an episode of game play. The weights on the player preference term and the author evaluation term in the heuristic function are adjusted depending on the "confidence" of the system that it has an accurate model of player preferences. Thus, if the system is able to confidently identify the current player as having a particular preference, it will guide her toward the types of stories she enjoys; otherwise, it will attempt to preserve author intent.

Several issues arise. First, the author's evaluation function must be defined over partial stories. Nelson & Mateas have previously discussed the difficulties in authoring evaluation functions that are well defined in this manner [93, 95]. Second, the particular choice of questions used for elicitation can be a cause for concern especially when the user is not completely sure of what she wants. Finally, it is unclear if the distinction between player preference models and player action models is necessary: explicitly modeling player preference may not provide increased representational power over implicitly modeling player preferences through the detailed modeling of their actions.

In any case, this system makes explicit the trade-off between **player self-agency** and **authorial control**. Further, the case-based approach is well-suited for online **adaption**. Of course, as with all learning techniques, CBR may require many examples to be effective, so extracting a player model may be difficult in practice. When it is difficult, the system reverts to SBDM. Further, when possible, the system cedes authorial control.

Evaluation of player preference models:

- **Speed**: ○, reliance on expectimax search and a growing case library can cause speed issues
- **Coordination**: ◐, this is dependent on the concrete implementation
- **Replayability**: ○, like OPIATE, as the case-base grows, the system's choices will stagnate and begin to rely heavily on the player for non-determinism
- **Authorial Control**: ○, the system will maximize for the player rather than the author if at all possible
- **Player Self-Agency**: ◐, this is dependent on the concrete implementation
- **Ease of Authoring**: ◐, like SBDM and DODM, authoring an abstract narrative seems easy but it's unclear if the evaluation function is feasible
- **Adaptability**: ●, designed to improve decision making in favor of the player's satisfaction
- **Soundness**: ○, no affordance provided for theoretical inquiry
- **Invisibility**: ◐, is dependent on the specific concrete implementation
- **Measurability**: ●, player satisfaction is measured through surveys

2.5.6 PaSSAGE

Thue *et al.* present the Player-Specific Stories via Automatically Generated Events (PaSSAGE) system [126, 127]. This system uses a three level hierarchy for defining a narrative similar to the idea of Mateas and Stern's *narrative sequencing*: the *event sequence* level where the components of the story are selected; the *structure* level where the details concerning the time and place of story events are determined; and lastly the behavior level where the actions of individual characters are determined. While each level of the specification is defined ahead of time by the game author, the library of available specifications is refined during game play to fit the individual player's characteristics.

The PaSSAGE system models the player's *style of play* in the game, refining its estimates as the narrative unfolds. The authors classify players according to five player types: fighters who prefer combat; power gamers who prefer gaining riches and items; tacticians who prefer thinking creatively; storytellers who prefer complex plots; and method actors who prefer dramatic actions. Based on the observation of the player's behavior in the game and annotations of plot events provided by the author, the system expresses its belief that the current player is of a specific type in the form of a weight vector. For example, if the system observes the player starting or joining an existing fight, it will increase the weight associated with the fighter player type.

Thus, similar to Barber and Kudenko's dilemma system, PaSSAGE manages the narrative experience by selecting among the set of story events that is most appealing to the currently estimated player weight values. Each event has a set of associated branches annotated with weights describing the appeal to each of the different player types. To determine the event and branch that is most appropriate, the inner-product is taken between the player weights and the author's weight annotations. The geometric interpretation of the inner-product is related to the cosine of the angle between the vectors. Thus, the more similar the vectors are, the higher the value of the inner-product will be.

This system excels in **speed** due to the simplicity of inner-product calculations. Additionally, the extensive use and refinement of a player model earns it high marks in the **adaptability** category; however, the exhaustive set of annotations required for the system to take advantage of this modeling results in significant **authorial burden**.

Evaluation of PaSSAGE:

- **Speed**: ●, the DM decision making process is determined by the calculation of an inner product between two small vectors
- **Coordination**: ◐, is based on the concrete implementation
- **Replayability**: ○, is dependent on the player as the sole source of non-determinism
- **Authorial Control**: ●, with concrete scripting and rich annotations this system provides significant authorial control
- **Player Self-Agency**: ●, constructs narratives in response to a player's decisions in the environment
- **Ease of Authoring**: ○, requires exhaustive and rich annotations of many sub-plots
- **Adaptability**: ●, maintains a model of player types based on observed game behavior and selects narrative events that fit well with specific player types
- **Soundness**: ○, no affordance for theoretical inquiry is provided
- **Invisibility**: ●, since the system generates rather than adapts narratives, it will be tough for players to identify the role of the DM
- **Measurability**: ○, no affordance for measurement is provided

2.6 Coordination Outside of Interactive Drama

Although we mainly discuss drama management systems in terms of interactive entertainment, we feel that efforts in applying such techniques in other domains are instructive. In this section, we briefly mention *narrative-based learning* and *game balancing*.

2.6.1 Narrative-based Learning

There has been growing interest in the use of games for instructional purposes. In educational and training environments, the teacher or trainer plays the role that the author plays in entertainment settings. Thus, the task of dynamically constructing engaging learning experiences in games is similar to the task of ensuring authorial intent in interactive narrative environments. Mott *et al.* have developed a multi-level planning architecture for narrative-based learning environments [87, 88, 89, 91]. Ultimately, the goal of their system is twofold. First, the system must support the hypothesis-generation-testing cycles that are the foundation of exploratory learning. Second, the system must provide appropriate levels of motivation and engagement for the learner to succeed.

Their system uses two *hierarchical task network* (HTN) planners that operate at two levels of abstraction. The *tutorial planner* constructs plans that reflect the educational goals of the teacher. On the other hand, the *narrative planner* determines how best to carry out the tutorial plans at the concrete game level. Tutorial plans constrain the plan space of the narrative plans.

Mott *et al.* describe their HTN-based system as providing an intuitive and **easy authorial idiom**; however, their deterministic planning approach reduces **replayability**.

In addition to the work of Mott *et al.*, Riedl *et al.* have also applied their work on ASD to training scenarios (see Section 5.3 for the discussion of that work).

Evaluation of Narrative-based learning:

- **Speed**: ○, as noted throughout this paper, planning is slow in any sizable domain
- **Coordination**: ◐, it is unclear if NPC agents make sense in this domain and is therefore author dependent
- **Replayability**: ○, the use of deterministic HTN planners requires that the player be the source of non-determinism
- **Authorial Control**: ●, is designed to guide players to a specific authorial goal
- **Player Self-Agency**: ●, designed to support exploratory learning

- **Ease of Authoring**: ●, the designers of this system describe it as providing an intuitive and easy authorial idiom

- **Adaptability**: ○, no model of player goals or preferences is included in the system

- **Soundness**: ○, no affordance is provided for theoretical inquiry

- **Invisibility**: ◐, is dependent on the set of actions provided to the DM

- **Measurability**: ○, no affordance is provided to measure educational goals of the teacher (or author)

2.6.2 Game Balancing

At a high level, drama management shares something in common with *dynamic game balancing*. That is, both game balancing agents and drama managers are tasked with making changes to the game world that will affect the player's experience. As discussed throughout this chapter, the drama manager is generally designed to ensure authorial intent; however, a game balancing agent tries to modify the game world to ensure maximal enjoyment by the player. In that sense, the work on player preference modeling, dilemma-based narratives, and PaSSAGE each have elements in common with game balancing approaches as well as drama management approaches.

A frequently discussed example of game balancing is that of a first person action game. The more frequently the game is played, the more skilled the player will become at the combinations of button presses and timing required to master the game. As the player's skill level increases, it is likely the game will become less challenging and potentially cause the player to lose interest; however, if the game's difficulty is adjusted to keep the player from mastering it, the player may also lose interest due to feeling like they are not improving. Traditionally, games have a static balancing component in the form of level selection (*e.g.*, easy, medium, hard, or expert). Recent AI research applied to game balancing has given rise to the field of dynamic game balancing where the traditional "discrete" balancing through explicit player selection is replaced with intelligent game adaption and replayability across game episodes.

Our treatment of dynamic game balancing here is brief as it is only peripherally related to our work; however, it is a rich area that supports a number of approaches, including reinforcement learning [1, 2, 3, 4, 5]; parameter manipulation [51]; dynamic scripting [120]; and genetic algorithms [31].

2.7 Discussion

The systems we have explored each have strengths; however, they all share common weaknesses. The approaches to drama management explored here have been focused on developing systems that provide some level of fidelity to the author's intent given a model of that intent; however, there is little evidence to suggest that any of the models proposed here are transparent to the typical author, who will presumably be an expert in narrative, but not in optimization, planning or any specific AI technique. The overarching goal behind the development of many of the techniques in this dissertation is to make the authoring process easier. We acknowledge that an author's ability to leverage our techniques to accomplish this goal has not been verified. See Section 8.4.1 for a more detailed discussion of this topic.

In this chapter, we have surveyed a variety of systems for drama management in interactive drama. We have proposed a number of desiderata, including **speed**, **coordination**, **replayability**, **authorial control**, **player self-agency**, **ease of authoring**, **adaptability**, **soundness**, **invisibility**, and **measurabilty**. We have described each of the systems we surveyed, including the work presented in this dissertation, in terms of these desiderata. In doing so, we have situated our techniques (TTD-MDPs, Section 2.3.3) in relation to other approaches.

CHAPTER III

TARGETED TRAJECTORY DISTRIBUTION MARKOV DECISION PROCESSES (TTD-MDPS)

In this chapter we discuss our solution to the goal selection problem for drama management (see Section 1.2). Our approach is based on modeling the problem as an online decision making problem. Here we will present the technical details of a formalism for describing and algorithms for solving an instance of a goal selection problem for drama management.

Markov Decision Processes (MDPs) are a popular formalism for modeling online decision making problems, particularly when the outcomes of decisions are only partially under the control of the decision maker. They were originally developed by Howard [49]. Typically, MDPs are solved using either dynamic programming or reinforcement learning [53]. MDPs have become very popular in various fields, with advances in theoretical artificial intelligence research as well as much more applied operations research.

In recent years, there has been been a significant increase in the popularity of MDPs in various research communities. There have been numerous generalizations to the original formalism. Those generalizations attempt to extend the basic MDP model in ways that make it a more realistic representation of the real world. For example, Partially Observable Markov Decision Processes (POMDPs) [54] ask the question: "What happens if the agent can't observe the real state of the world?" POMDPs provide an answer to this question by introducing the concept of observation vectors and the probabilistic relationship between the observed state and the true state. In Semi-Markov Decision Processes (SMDPs) [50] the question is asked: "What if some of the agent's actions aren't instantaneous?" SMDPs provide an answer to this question by introducing a temporal model of action duration. Another example is Hard-Constrained MDPs [138] where there are costs associated with actions and a hard constraint on the cumulative cost cannot be exceeded (for example a robot cannot run on an empty battery). All of these extensions to MDPs, as well as others, add complexity to the model by relaxing one or more of the simplifying assumptions made in the original MDP model. Despite the added complexity, all of these models retain the original goal of an MDP which is to provide a framework for reasoning about how sequences of decisions affect an agent's ability to "optimally" realize its goal.

In this chapter, we describe a new variant of MDPs known as ***Targeted Trajectory Distribution MDPs*** or TTD-MDPs. Rather than make a novel complexity-increasing relaxation of the traditional MDP model, we ask a different question: "What happens if the path the agent takes to the goal is more important than simply getting there?"—a question inspired by the DODM formalism. TTD-MDPs are a novel class of decision problems where the decisions that are made are intended to optimize behavior in the limit of infinite repetition of episodes—traditionally "optimal" behavior during any given episode is sacrificed in order to achieve a desired target distribution over behaviors. TTD-MDPs were originally developed to enable authors of interactive dramas to design experiences that were unpredictable to the player in a way that was predictable to the author [112]. Enabling this *predictable unpredictability* is the major benefit of TTD-MDPs over traditional MDPs.

We note that adding nondeterminism to MDP policies is not new. Other systems such as Markov Games use non-deterministic policies to get "optimal" behavior in a two-player adversarial setting [64]. Again, the goal of that formalism to fix a policy that will result in the highest payoff for the agent. On the other hand, the Cobot system uses nondeterministic policies for an agent that interacts with users of an online virtual community [52]. In an effort to prevent Cobot from being overly repetitive and therefore less interesting for users to interact with, Cobot took actions in proportion to their estimated value. While this local variation served Cobot well, its effect on the complete experience is not well understood. Learning to optimize reward in an MDP and then adding some *ad hoc* nondeterminism does not allow the agent to reason about how local decisions have an effect on the complete sequence of decisions.

In this chapter we describe TTD-MDPs in detail and highlight the following contributions of our work:

- The relationship between an MDP and a TTD-MDP
- How to convert an MDP into a TTD-MDP
- How to author a TTD-MDP directly
- Solving a TTD-MDP using a provably correct optimization-based method
- Strategies for scalable TTD-MDP policy computations
- Two authorial idoms for target distributions

In addition to the above, we will recount how the problem of managing experiences in interactive entertainment motivated the development of TTD-MDPs.

3.1 From MDPs to TTD-MDPs: A Formal Definition

In this section we will present a formal definition of TTD-MDPs. In the next section, we will explain each of the components in more detail. To be consistent with notation from the literature, P will be used generically to define a probability distribution in various contexts. The reader should be careful to discern the specific distribution that P refers to which should be clear from the context.

A typical MDP is defined by a tuple $(\mathcal{S}, \mathcal{A}, P, R)$, where $\mathcal{S}$ is a set of states, $\mathcal{A}$ is a set of actions, $P : \{\mathcal{S} \times \mathcal{A} \times \mathcal{S}\} \rightarrow [0, 1]$ is a probability distribution defining transitions between state given actions, and $R : \mathcal{S} \rightarrow \mathbf{R}$ is a reward function. The solution to an MDP is a policy $\pi : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$. The policy specifies a probability distribution over action choices in the given state. An optimal policy ensures that the agent receives maximal long-term expected reward.

A TTD-MDP shares components with a traditional MDP. Specifically, a TTD-MDP is defined by a tuple $(\mathcal{T}, \mathcal{A}, P, P(\mathcal{T}))$, with trajectories $\mathcal{T}$ that are partial or complete trajectories of MDP states; a set of actions $\mathcal{A}$; a transition model $P : \{\mathcal{T} \times \mathcal{A} \times \mathcal{T}\} \rightarrow [0, 1]$ is a probability distribution defining transitions between trajectories given actions; and a target distribution over complete trajectories $P(\mathcal{T})$ ($P(t)$ will refer to the target probability of a given partial trajectory and $P(\tau)$ will refer to the target probability of a given complete trajectory).[1] We will sometimes use the notation $P(t'|a, t)$ to refer to the probability of t' given a and t. Note that the target distribution of a TTD-MDP replaces the reward function of a traditional MDP. The solution to a TTD-MDP is a policy $\pi : \mathcal{T} \times A \rightarrow [0, 1]$ providing a distribution over actions in every partial trajectory. Additionally, we will sometimes use the notation $P(a|t)$ to refer to the policy $\pi(t, a)$. The optimal policy is defined to be the policy that results in long-term behavior under repeated episodes as "close" to the target distribution as possible.

Any discrete-time MDP can be converted to a TTD-MDP. Consider an MDP with a set of states $\mathcal{S}$ and sets of actions available in each state $\mathcal{A}_s$, the probability $P_{i+1}(s')$ that the process is in state s' at time $i + 1$ is defined recursively by:

$$P_{i+1}(s') = \sum_{\forall s \in \mathcal{S}, a \in \mathcal{A}_s} (P(s'|a, s) \cdot P(a|s) \cdot P_i(s)) \tag{1}$$

where $P(s'|a, s)$ is the transition model encoding the dynamics of the world and $P(a|s)$ is the policy under the agent's control. During an actual episode, $P_i(s) = 1$; if we assume (as is commonly done) that the policy is deterministic, we get a common form of Equation 1, rewritten as: $P_{i+1}(s') = \sum_{\forall s \in \mathcal{S}, a \in \mathcal{A}_s} P(s'|\pi(s), s)$.

Because we are interested in trajectories in TTD-MDPs, we can simply roll the history of the MDP states into the TTD-MDP trajectories, resulting in a TTD-MDP where each trajectory represents a sequence of states in the underlying MDP, optionally including a history of the actions taken.

[1] Optionally including actions in the trajectory, if necessary, allows us to ensure that our TTD-MDP is not underconstrained.

Dealing with trajectories means that the "state" space of the TTD-MDP forms a tree. The power of this insight becomes evident when we restate Equation 1 for TTD-MDPs:

$$P(t') = \sum_{\forall a \in \mathcal{A}_t} (P(t'|a,t) \cdot P(a|t)) \cdot P(t). \tag{2}$$

In other words, for every partial or full trajectory t', the transition probability $P(t'|a,t)$ is nonzero for exactly one $t \preceq t'$ that is its immediate prefix. This observation follows from the fact that each trajectory represents a unique sequence of states $s_1, a_1, \ldots, a_{\|t\|-1}, s_{\|t\|}$ and therefore has a unique prefix. Thus, the summation need only account for possible actions taken in the preceding partial trajectory rather than actions in multiple MDP states. Because each trajectory has a fixed length and can therefore appear at only one specific time, we can drop the i subscripts for time.

Finally, we need a target distribution. In general, any arbitrary target distribution can be used. There are a variety of ways one might imagine encoding a distribution. For now we will assume a distribution has been specified and may be queried. Later, in Section 3.4, we will discuss techniques to author target distributions TTD-MDPs.

3.2 Components of a TTD-MDP

In this section, we will describe in detail the various components of a TTD-MDP expanding upon the formal definition presented in the previous section. The purpose of this section is to try to make the reader's understanding of TTD-MDPs more precise as well as illustrate the broader applicability of TTD-MDPs.

Briefly, a TTD-MDP is a decision process that differs from traditional decision processes in two ways: first, decisions are made not just based on the state of the environment, but how the agent came to that state (*e.g.*, the Markov assumption does not apply); and second, decisions are made probabilistically to obtain desirable behavior under repetition, not just during a single episode. A TTD-MDP has four components which we will describe in more detail below:

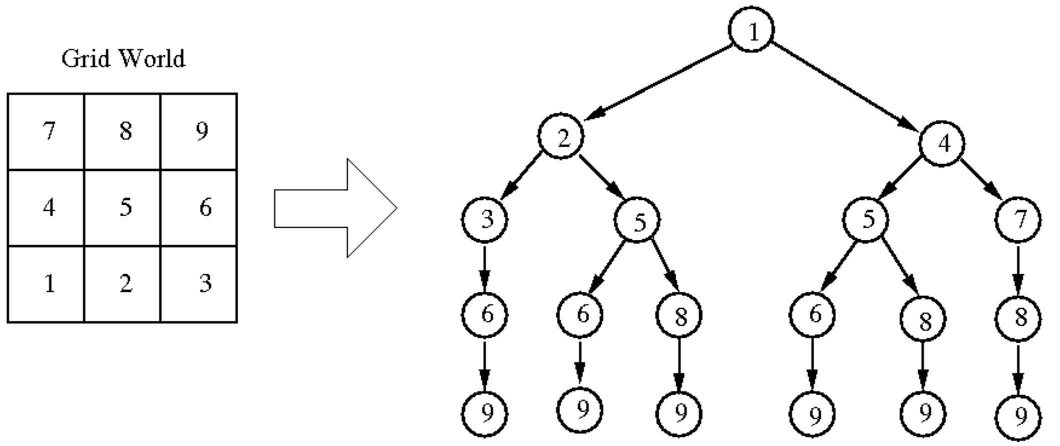


Figure 3: An example 3×3 gridworld with *right* and *up* actions and the resulting trajectory tree.

- A set, $\mathcal{T}$, of partial (t) and complete (τ) trajectories
- A set, A, of actions applicable in trajectories
- A model of the system dynamics, $P(t'|a,t)$, describing (possibly probabilistic) transitions between trajectories
- A target distribution, $P(\tau)$, describing desired probabilities of complete and, by implication, partial trajectories as well

The solution to a TTD-MDP, just like any other type of decision process, is a policy describing action choices in every state.

3.2.1 Trajectories

Trajectories represent total history traces of an online decision making process. In Figure 3, we illustrate in detail what a trajectory is using an example 3×3 gridworld. We consider the case where there are two deterministic actions: *move right* in the grid (R) and *move up* in the grid (U). The left side of the figure depicts the grid world and the right side of the figure depicts a *trajectory tree*. A trajectory tree is simply a graphical representation of the set of valid trajectories and the prefix relationship that governs partial trajectories.

All legal trajectories have initial state 1 and terminating state 9. For instance, consider the partial trajectory $1 \xrightarrow{R} 2$ representing the agent having started in state 1, taken action R, and moved to state 2. It is a prefix of two immediate subsequent partial trajectories ($1 \xrightarrow{R} 2 \xrightarrow{R} 3$ and $1 \xrightarrow{R} 2 \xrightarrow{U} 5$) as well as three other partial trajectories and three complete trajectories.

The important point of this example is that although "states" 5, 6, 8, and 9 all appear in the trajectory tree many times, they actually represent different trajectories depending on the path taken to arrive at that state. For simplicity, we will look at both of the appearances of state 5. The one on the left of the tree in Figure 3 represents the partial trajectory $1 \xrightarrow{R} 2 \xrightarrow{U} 5$ whereas the one on the right represents the partial trajectory $1 \xrightarrow{U} 4 \xrightarrow{R} 5$—they depict the same world *state*, but they are different *trajectories* because the process arrived in that state along a different path.

In a narrative setting, the path from the beginning to the end of the story is just as important, if not more so, than the ending itself. Efficiency isn't necessarily the desirable quality of a good path through a story. Further, once a story event has occurred, it cannot be undone, nor can it occur again. Thus, for storytelling, we are interested primarily in trajectories. The power of considering trajectories rather than states is great. An agent, such as a drama manager, can not only make decisions based on where it is but also based on how it got there. It does, however, introduce certain complexities we will discuss in later sections.

Suppose we have an agent that acts as a personal trainer and in doing so instructs joggers on what path to take through a park. Using trajectories instead of states, the decision maker can guide the jogger in more complicated patterns like two laps around a circle or a figure eight pattern. Using a simple state representation precludes this type of decision making.

3.2.2 Target Distribution vs. Reward

In any decision process, it must be the case that the agent making the decisions is working toward some goal. Traditionally, in MDPs that goal has been to maximize some reward function. The reward "signal" is passed to the agent every time they enter a state. When the agent learns a policy, it explores the world, keeping track of the reward it receives in each state. In doing so, it maintains an estimate of *utility* which is traditionally defined as the expected discounted reward for each state. After sufficient time, when the agent has constructed a reasonable estimate of the utility of each of the states, its policy is implicitly defined by those estimates.

In some cases, the reward function can represent the outcome of a real world process (like a game such as chess or Othello) and in others it can be completely hand authored in an attempt to bring about specific behavior (as is typically the case in controllers for robots). For example, in a simple non-deterministic grid-world domain, with a start state and two terminal states—one with positive reward 1.0 and one with negative reward -1.0—we can control the behavior of the decision making agent by selecting the magnitude of the intermediate reward accordingly. If we wish the agent to be risk averse (*e.g.*, we wish they avoid the terminal state with negative reward as frequently as possible) we can set the intermediate reward to something small and positive. On the other hand, if we wish they be risk tolerant (*e.g.*, we wish the agent move to a terminal state as quickly as possible regardless of which it is) we will set the intermediate reward to relatively large and negative. In this manner, by careful construction of a reward function we can exert much control over the characteristics of the decision maker's policy.

While this control available to an MDP designer is good, it is not necessarily easy to take advantage of. Specifically, in large domains with hundreds or thousands of states, it is difficult to hand author a reward function that will result in exactly the behavior desired. Further, there is one serious shortcoming with the traditional reward-based paradigm for decision processes—there is no known method to enforce a *principled* non-determinism in the decision maker's policy.

In TTD-MDPs, we have all of the power to control the characteristics of the decision maker's policy and more. This power, however, comes not through a reward function, but a target distribution. The idea of a target distribution relies on the notion of trajectory discussed in Section 3.2.1. Rather than pass a reward to the agent upon entering a state, we tell the agent the relative desired probability of all subsequent states and allow them to probabilistically choose an action that will—in the limit of infinitely many repeated episodes—arrive at each of the subsequent states the correct proportion of the time. Note that in this paradigm, we can engineer similarly risk tolerant or risk averse behavior. In the same simple grid-world discussed above, if we have a target distribution that is closer to uniform, the agent will be risk tolerant. If the target distribution is more kurtotic with peaks over trajectories that end up in the positive terminal state, the agent will be risk averse. The authoring of target distributions has the potential to be easier than authoring of reward functions for similar behavior.

Typically, we consider a target distribution that gives probability mass to complete trajectories only. A partial trajectory is a sequence of world states that does not end in a terminal state. One of the challenges with TTD-MDPs is the need to know the target probability for partial trajectories even though the target distribution may only be defined over complete trajectories. In order for an agent to realize the desired distribution over complete trajectories, it must know the relative probability mass for each of the partial trajectories reachable from the partial trajectory it is making the action choice in.

3.2.3 Transitions

This section is devoted to a discussion of $P(t'|a,t)$—the probability of a particular trajectory t' being realized given the process is in another trajectory t and the action a has been taken. This is related to the transition model from traditional MDPs and governs the dynamics of the TTD-MDP. There are a few things to note about this particular distribution:

- If there are deterministic transitions, then $P(t'|a,t) \neq 0$ for exactly one t'
- $P(t'|a,t) \neq 0$ for only those trajectories t' such that t is an immediate prefix, *i.e.*, the difference between trajectories t' and t is one event that occurs subsequent to t:

$t' = t : e_n$

These two characteristics are not significantly different from the transition dynamics of a traditional MDP. For example, the transition function $P(s'|a, s)$ in a traditional MDP is non-zero for exactly the set of states s' that are reachable in one step from the current state s; however, looking in reverse, the set of states s that enable an agent to reach s' can also be larger than one. In other words, the transition model for MDPs is Many-to-Many whereas in TTD-MDPs, due to the nature of trajectories, the transition model is One-to-Many.

While this difference in transition models may seem subtle, there is notable power that comes from this property. This One-to-Many characteristic simplifies the computation of the TTD-MDP policy significantly by constraining the set of valid prefix trajectories to only one. Additionally, it is this property that enables the efficient solution of a TTD-MDP. This will be discussed in more detail in Section 3.3.

3.2.4 Policies

The solution to a TTD-MDP, like a traditional MDP, is a policy mapping states to action choices. In almost all work on MDPs, the form of this policy is deterministic—for each state there is an "optimal" choice of action. This deterministic policy function, $\pi : S \to A$, provides a single choice of action in any given state. The application of this policy during an episode will bring behavior intended to maximize the long-term reward received from the MDP reward function. Note, however, that non-determinism in the environment will bring about some variation in realized trajectories during repeated episodes. This non-determinism is beyond the control of the decision making agent and, in situations where the MDP models a real world environment, is generally beyond the control of the MDP designer as well.

The goal of solving a TTD-MDP is to fix a policy that will result in longer term "optimal" behavior. Here, the definition of optimal does not pertain to maximizing reward (as there is no reward in a TTD-MDP). The definition of optimal pertains to the distribution of trajectories realized after repeated episodes of the decision making process. In other words, rather than determine an optimal action at every point during any given episode, the policy should provide a distribution over actions that when repeatedly applied will result in a set of trajectories that matches the provided target distribution as "closely" as possible. Therefore, in the TTD-MDP paradigm, the policy is a function $\pi : \mathcal{T} \times A \rightarrow [0, 1]$. Thus, for any $t \in \mathcal{T}$, we have $\sum_{a \in A} \pi(t, a) = 1.0$. The use of a non-deterministic policy allows for the decision making agent to add some control to the non-determinism in the system.

One might ask, why not use a standard MDP framework but take actions in proportion to the expected discounted reward received for taking them? The answer to this is twofold. First, the optimization methods for solving an MDP (*e.g.*, dynamic programming, reinforcement learning, etc.) either don't scale well to large environments or are primarily concerned with obtaining ordinally correct values for actions rather than numerically accurate estimates. It is important to note the subtle difference between ordinally correct values and numerically accurate estimates. When choosing actions to take simply by examining the expected value associated with them, using an $\operatorname{argmax}_{a \in A}$ which returns the proper action based on the estimate requires that the estimated values of the actions be correct in comparative magnitude only. On the other hand, when using estimated values to obtain a target distribution over action choices, the values must be accurate in scale as well. Specifically, if $V(a)$ is the true value and $V'(a)$ is the estimated value, then $\forall a \in A$ it must be that $V'(a) = c \cdot V(a)$ for some constant c. Second, in order for the relative values of actions to give the right probabilistic behavior in the limit of repeated episodes, the reward function would have to be authored so the environment's dynamics will properly distribute the reward values to each state in such a manner that an accurate distribution can be obtained.

For example, consider the following situation represented in Figure 4. An agent is in a particular partial trajectory t with two actions available, a_1 and a_2, that lead to two subsequent trajectories, t_1 and t_2. Further, suppose that the transition dynamics are such that $P(t_1|a_1, t) = 0.3$, $P(t_1|a_2, t) = 0.6$, $P(t_2|a_1, t) = 0.7$, and $P(t_2|a_2, t) = 0.4$ and the

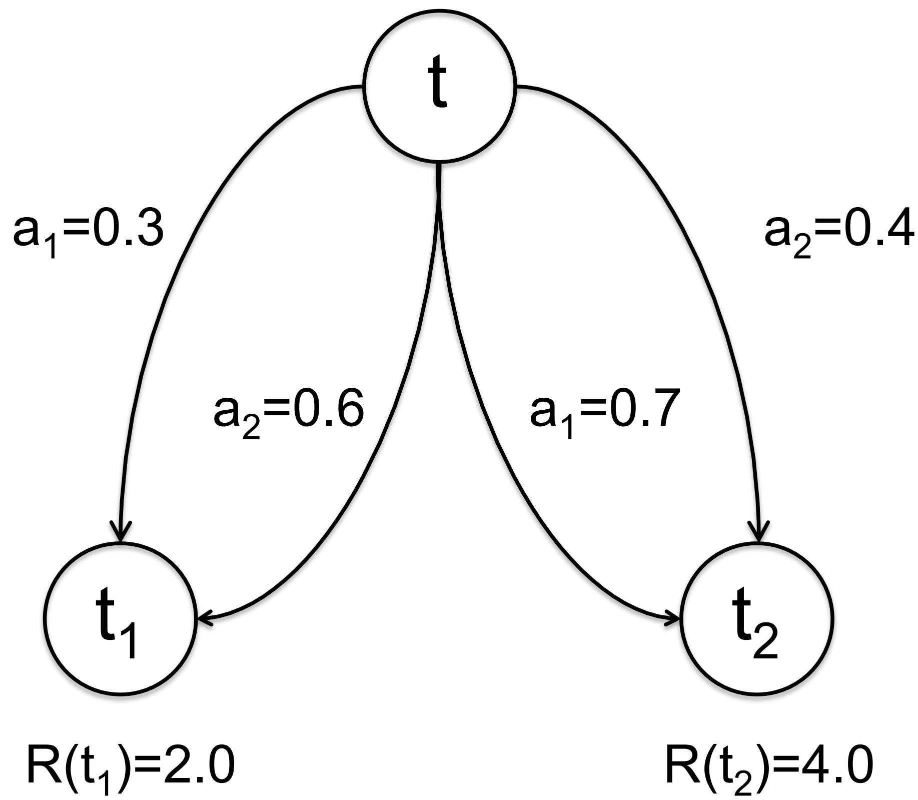


Figure 4: A small sample to illustrate reward backup and how it affects probabilistic action choice.

rewards the agent receives are as follows: $R(t_1) = 2.0$ and $R(t_2) = 4.0$. Thus, any MDP solution technique that attributes a higher utility estimate to a_1 than to a_2 will result in a policy that maximizes long term reward. Note, there are infinitely many estimates such that $V(t, a_1) > V(t, a_2)$. Now, for comparison, assume that the rewards are actually an indicator of relative goodness. That is, we wish to end up in t_1 $\frac{1}{3} = \frac{2.0}{2.0+4.0}$ of the time and in t_2 $\frac{2}{3} = \frac{4.0}{2.0+4.0}$ of the time. A little algebra (which will be covered in detail in Section 3.3) indicates that action a_1 should be taken with probability $\frac{8}{9}$ and action a_2 should be taken with probability $\frac{1}{9}$. If the agent is using utility estimates as the basis for its decision on action choices, then it must be the case that $V(a_1) = 8 \times V(a_2)$; while there are infinitely many utility estimates such that this would be true as well, it is the case that it is nearly impossible in practice to observe this behavior. Although we won't provide results, we have found performance of agents that act according to this approach to be poor at best.

3.3 Solving a TTD-MDP

We now define an algorithm to compute a policy $P(a|t)$ for every partial trajectory in $\mathcal{T}$. The basic strategy of this algorithm is to solve for $P(a|t)$ following a post-order traversal of the decision points along each of the trajectories. As mentioned above, one of the major benefits to working with trajectories over states is that the space of decision points forms a tree. In this algorithm, we exploit that fact.

Since we are interested in a conditional distribution, the problem of finding $P(a|t)$ reduces to performing a set of local computations for each trajectory and the trajectories immediately subsequent to it. The solution presented in Algorithm 1 is formulated so these local computations take the form of solving a system of linear equations. In order to have a well specified linear system it is necessary for the number of trajectories reachable from the current trajectory to be at least as large as the number of available actions. If this is not the case, the action taken can be included in the representation of the trajectory so the same state-based trajectory will appear to be different depending on the action taken (as in the example from Figure 3).

Algorithm 1 Algorithm to solve for π.

1: Build a tree of all possible trajectories.
2: Initialize each leaf node (complete trajectory) with its target probability $P(\tau)$.
   In reverse topological order:
3: **for** Every t **do**
4:     **for** Every child t'_i of trajectory t **do**
5:         Condition Equation 2 on t:

$$P(t'_i|t) = \sum_{\forall a \in \mathcal{A}_t} (P(t'_i|a, t) \cdot P(a|t))$$

6:     **end for**
7:     This forms a system of linear equations:

$$\vec{P}(t'_i|t) = \vec{P}(t'_i|a, t) \cdot \vec{\pi}(t)$$

   which can be solved for π.
8: **end for**

Algorithm 1 provides the basic process for solving a TTD-MDP. The notation used in Step 7, namely $\vec{P}$ and $\vec{\pi}$ is used to indicate either a vector or matrix. Below, an example will make this more clear. The significant details of the algorithm, particularly in Step 7, have been omitted. In the remainder of Section 3.3 we will present the details of this basic algorithm. We have developed three methods to solve Step 7 of Algorithm 1 [15, 112]. The first is a fast linear-algebra approximation (Section 3.3.1), the second is a convex optimization method based on minimizing local L_1 error (Section 3.3.2), and the third is a provably optimal convex optimization method that minimizes global error measured by KL divergence (Section 3.3.3).

3.3.1 A Fast Linear Algebra Approximation

Consider the equation presented in Step 7 of Algorithm 1. Each of $\vec{P}(t'_i|t)$, $\vec{P}(t'_i|a, t)$ and $\vec{\pi}(t)$ are matrices that represent the complete dynamics of a local decision point. Assume we are in a situation with current trajectory t, three possible subsequent trajectories t_1, t_2, and t_3, and three possible actions a_1, a_2, and a_3. Then we have:

$$\vec{P}(t'_i|t) = \begin{bmatrix} P(t_1|t) \\ P(t_2|t) \\ P(t_3|t) \end{bmatrix} \tag{3}$$

$$\vec{P}(t'_i|a,t) = \begin{bmatrix} P(t_1|a_1,t) & P(t_1|a_2,t) & P(t_1|a_3,t) \\ P(t_2|a_1,t) & P(t_2|a_2,t) & P(t_2|a_3,t) \\ P(t_3|a_1,t) & P(t_3|a_2,t) & P(t_3|a_3,t) \end{bmatrix} \tag{4}$$

$$\vec{\pi}(t) = \begin{bmatrix} P(a_1|t) \\ P(a_2|t) \\ P(a_3|t) \end{bmatrix} \tag{5}$$

and therefore:

$$\begin{bmatrix} P(t_1|t) \\ P(t_2|t) \\ P(t_3|t) \end{bmatrix} = \begin{bmatrix} P(t_1|a_1,t) & P(t_1|a_2,t) & P(t_1|a_3,t) \\ P(t_2|a_1,t) & P(t_2|a_2,t) & P(t_2|a_3,t) \\ P(t_3|a_1,t) & P(t_3|a_2,t) & P(t_3|a_3,t) \end{bmatrix} \cdot \begin{bmatrix} P(a_1|t) \\ P(a_2|t) \\ P(a_3|t) \end{bmatrix} \tag{6}$$

For simplicity, we will write Equation 6 as $\vec{y} = \vec{A} \cdot \vec{x}$ where $\vec{y}$ is the vector of target probabilities for each of the three subsequent trajectories, $\vec{A}$ is the matrix of transition probabilities, and $\vec{x}$ is the vector of action probabilities (or policy). $\vec{y}$ and $\vec{A}$ are specified as part of the TTD-MDP model, $\vec{x}$ is under our control and is to be solved for. Thus, the algebraic solution: $\vec{x} = \vec{A}^{-1} \cdot \vec{y}$ is what we are looking for; however, $\vec{A}$ need not have an inverse, so in practice we use the "pseudoinverse" of $\vec{A}$, which results in:

$$\vec{x} = (\vec{A}^T \cdot \vec{A})^{-1} \cdot \vec{A}^T \cdot \vec{y} \tag{7}$$

This procedure is guaranteed to minimize the error vector $\|\vec{A} \cdot \vec{x} - \vec{y}\|_2$.

While this approach seems elegant, unfortunately in practice there are often situations where TTD-MDP policy computations do not work out nicely. Here, we will describe two types of complications that arise during solving a TTD-MDP and provide an alternate solution technique that overcomes these limitations.

3.3.2 Impossible Constraints and L_1 Optimality

Empirically, we find this approach almost always results in an optimal policy; however, this is not always possible. There are two types of errors we can encounter. First, there may be no vector $\vec{\pi}(t)$ that satisfies the linear system exactly. Second, even when there is an exact solution, the elements of $\vec{\pi}(t)$ may not be probabilities (though they will still sum to 1.0).

Lemma 1. *For a given trajectory t with subsequent trajectories $t'_1, \ldots, t'_n$ and actions $a_1, \ldots, a_m$, the following condition is sufficient for there to be no exact solution to the system of equations where the entries of $\vec{\pi}(t)$ are probabilities:*

$$\exists i \textit{ such that either}$$

$$\forall a : P(t'_i | a, t) < P(t'_i | t) \tag{8}$$

$$\textit{or } \forall a : P(t'_i | a, t) > P(t'_i | t) \tag{9}$$

Proof. The proof follows from the observation that unless the probabilities of two actions bracket the desired distribution (or at least one matches exactly) there is no convex combination of actions that can result in the desired distribution. □

Corollary 2. *If the system has an exact solution, then for every t'_i, there exist a_j, a_k such that $P(t'_i | a_j, t) \leq P(t'_i | t)$ and $P(t'_i | a_k, t) \geq P(t'_i | t)$.*

For example, consider a trajectory t', three subsequent trajectories t_1, t_2, t_3 and three actions a_1, a_2, a_3 whose linear system is defined by:

$$\begin{bmatrix} 0.0 \\ 0.3333 \\ 0.6667 \end{bmatrix} = \begin{bmatrix} 0.5 & 0.5 & 0.0 \\ 0.0 & 0.5 & 0.5 \\ 0.5 & 0.0 & 0.5 \end{bmatrix} \cdot \vec{\pi}(t) \tag{10}$$

Then, the solution vector

$$\vec{\pi}(t) = \begin{bmatrix} 0.3333 \\ -0.3333 \\ 1.0000 \end{bmatrix}$$

does not represent a probability distribution.

While this solution is not a vector of probabilities, it does satisfy the linear system. Intuitively, achieving the desired distribution requires that action a_2 be "undone" some percentage of the time. Since it is impossible, in practice we zero out any negative values and re-normalize. To handle these cases where an exact solution is either not possible or is not a valid probability distribution, we define $\widehat{\pi}(t)$ whose entries are based on $\vec{\pi}(t)$ in the following way: Let $\mathbf{O} = \{j : \vec{\pi}(a_j|t) < 0\}$ then

$$\widehat{\pi}(t) = \left[\begin{Bmatrix} 0.0 & \text{if } i \in \mathbf{O} \\ \frac{\vec{\pi}(a_i|t)}{\sum_{j \notin \mathbf{O}} \vec{\pi}(a_j|t)} & \text{otherwise} \end{Bmatrix} \right] \tag{11}$$

Returning to the example from above, calculating $\vec{P}(t'_i|t) = \vec{P}(t'_i|a,t) \cdot \widehat{\pi}(t)$, we get the following:

$$\widehat{\pi}(t) = \begin{bmatrix} 0.25 \\ 0.0 \\ 0.75 \end{bmatrix} \text{ and } \vec{P}(t'_i|a,t) \cdot \widehat{\pi}(t) = \begin{bmatrix} 0.1250 \\ 0.3750 \\ 0.5000 \end{bmatrix}$$

where we can calculate $\|\vec{P}(t'_i|a,t) \cdot \widehat{\pi}(t) - \vec{P}(t'_i|t)\|_1 = 0.3333$.

We have derived a lower bound on L_1 error when Lemma 1 holds for only one trajectory t_i:

$$\|\vec{P}(t'_i|a,t) \cdot \widehat{\pi}(t) - \vec{P}(t'_i|t)\|_1 \geq 2 \times \left[\min_a |P(t'_i|a,t) - P(t'_i|t)|\right]$$

. There are cases where this procedure does not obtain this lower bound. In Section 6.3 we will present the results of experiments that indicate how infrequent this occurrence is in practice.

Although this procedure works well in practice, it is not guaranteed to minimize error. Figure 5 illustrates why. These figures provide a geometric interpretation of the normalization step. The solution space to the linear system lies somewhere on a hyperplane constrained by the transition matrix. Because we want the solution to be probabilities, the region of valid solutions on this hyperplane must lie somewhere between 0 and 1 on each

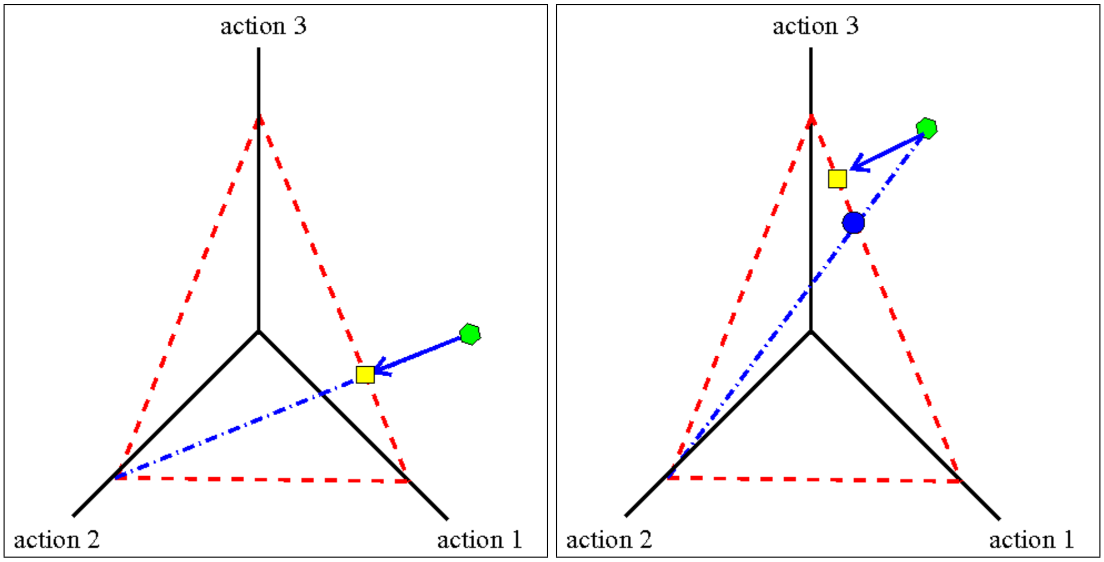


Figure 5: Geometric view of the normalization procedure in a problem instance with three actions.

of the axes (represented by the dashed triangle connecting the axes). In the case presented in Equation 10, we have a situation where the probability restriction is not met. This is depicted in the figures by the hexagon farthest to the right. In Figure 5(a) we depict a situation where symmetry in the transition matrix results in an optimal solution and in Figure 5(b) we depict a situation where this symmetry does not exist. The normalization procedure is represented by the dashed line from the solution dot (hexagon on the far right) to the axis for the action that received negative mass (axis on the far left). In Figure 5(b) we have emphasized where this line crossed the boundary of the hyperplane by including a circle. The intuition here is that by assigning zero mass to the action(s) whose solution is negative and normalizing we are maintaining the probabilities of the other actions in the same relative proportion. On the other hand, the optimal solution is the point in the valid region of the hyperplane that is closest to the actual solution (*i.e.*, lies on a perpendicular line from the boundary of the valid region to the solution point). This optimal result is depicted by the square that lies on the hyperplane boundary at the end of the arrow. Note that in the symmetric case, the location of the optimal solution and the normalized solution are the same.

In the asymmetric case, the location of the normalized solution does not coincide with the optimal solution.

To achieve true local optimality, it is necessary to solve a constrained optimization of the L_1 error. The objective function is:

$$\min_{\vec{\pi}} \|\vec{P}(t'_i|t) - \vec{P}(t'_i|a,t) \cdot \vec{\pi}\|_1 \tag{12}$$

subject to $\sum_a \pi(t,a) = 1$ and $0 \leq \pi(t,a) \leq 1$ [15]. Standard techniques exist to reduce this objective function to a constrained linear program [16]. This optimization procedure replaces the matrix computation presented in Section 3.3.1 for Step 7 of Algorithm 1 and is guaranteed to minimize *local* L_1 error.

3.3.3 A KL Optimal Approach

In this section, we present an algorithm for solving TTD-MDPs based on the *Kullback Liebler divergence*. The KL-divergence [61] between two probability distributions $p(x)$ and $q(x)$ is defined as:

$$D_{KL}(p\|q) = \sum_x p(x) \log \frac{p(x)}{q(x)} \tag{13}$$

$$= \sum_x p(x) \log p(x) - \sum_x p(x) \log q(x) \tag{14}$$

KL-divergence is not a true distance, as it is asymmetric; however, it is a well-understood measure with several important properties. In particular, it is consistent, always non-negative and zero only when p and q are equal. If we think of p as a base distribution then D_{KL} measures how well q approximates it, by measuring the entropy that remains in q when p is known. In context of TTD-MDPs, p is the target distribution and q is the approximation of it obtained by combining the world dynamics (transition model) with the probabilistic policy:

$$q(\tau) = \prod_{t \preceq \tau} w(t) \tag{15}$$

$$\text{where} \quad w(t') = \sum_a P(t'|a,t) \cdot \pi(t,a) \tag{16}$$

Here, τ represents complete trajectories while t and t' are (partial or complete) trajectories. The probability $w(t')$ represents the frequency with which t' is targeted when the process is at t. It combines information about the probabilistic policy and world dynamics at t. Our objective function for optimization then becomes:

$$\begin{aligned}\max_{\pi}\sum_{\tau}p(\tau)\log q(\tau) &= \max_{\pi}\sum_{\tau}p(\tau)\log\prod_{t\preceq\tau}w(t)\\ &= \max_{\pi}\sum_{\tau}p(\tau)\sum_{t\preceq\tau}\log w(t)\\ &= \max_{\pi}\sum_{\tau}\sum_{t\preceq\tau}p(\tau)\log w(t)\end{aligned}$$

A partial trajectory t contributes $p(\tau)\log w(t)$ to the sum for each complete trajectory τ for which it is a prefix. We can define a function over complete trajectories summarizing the factor of $\log w(t)$ that t contributes, $m(t) = \sum_{\tau,t\preceq\tau} p(\tau)$. Our objective function is then:

$$\max_{\pi}\sum_{t}m(t)\log w(t) \tag{17}$$

Note that $m(t)$ represents the total probability mass contained in the subtree rooted at t. Obtaining the optimal policy is simply a matter of performing an $\operatorname{argmax}_{\pi}$ on the objective function.

Having summarized and isolated the contribution to the objective of an individual trajectory t', we could proceed using a naïve approach and optimize for each trajectory independently, thus optimizing their sum; however, this procedure ignores the fact that optimizing for one trajectory may come at the cost of sacrificing the optimality of a sibling trajectory. Fortunately, optimizing for a trajectory t' is only constrained by the optimization of its sibling trajectories, and no others. This insight enables us to consider trajectories by groups of siblings. Fortunately, we have two distinct advantages: 1) the groups of siblings can be solved for in *any* order (there is no restriction that you must start with the leaves and work towards the root); and 2) solving the local optimizations is guaranteed to produce the globally optimal setting of the parameters. The local optimization is

$$\operatorname*{argmax}_{\pi_t} \left[\sum_{t \to t'} m(t') \log w(t') \right] \tag{18}$$

$$= \operatorname*{argmax}_{\pi_t} \left[\sum_{t \to t'} m(t') \log \sum_a P(t'|a,t) \cdot \pi_t(a) \right] \tag{19}$$

where we have used $t \to t'$ to indicate that t is a prefix (not necessarily immediate) of t'. This objective is convex, so Equation 18 can be solved using a standard technique for constrained convex optimization [16].[2] Thus, we yield a KL-optimal offline algorithm by solving Step 7 of Algorithm 1 using this approach.

One potential problem arises when $q(\tau)$ is forced to be zero for a complete trajectory τ with $p(\tau) \neq 0$. This occurs when no actions available at a trajectory t will get us to a child t', *i.e.*, $w(t') = 0$. $D_{KL}(p\|q)$ is undefined due to the division by zero. This has the effect of treating all possible approximations as equally bad, preventing us from making progress towards a solution. Luckily, this problem can be eliminated by preprocessing and reformulating each local optimization to eliminate child trajectories t' that can never be reached. Intuitively, t' should not be represented in the trajectory tree if it cannot ever be reached from t.

3.3.4 Intractable Problems

In practice, the tree that represents all of the trajectories may be infeasible to compute and store, or completely specifying a distribution over all trajectories may be difficult or impossible. Here, we introduce a method for sampling trajectories from the underlying MDP dynamics to handle this case. The basic strategy is to sample some number of trajectories from $P(\mathcal{T})$ and build a smaller trajectory tree.

The number of trajectories to be sampled can be determined by memory size or the

[2] Complete details are beyond the scope of this dissertation, but to solve the optimization step for KL-divergence, we implemented a log barrier method. It is provably convergent for the optimization of convex objective functions with linear inequality constraints [16]. This approach involves a convex objective function, so the solution returned by the constrained optimization routine is indeed optimal with respect to the local objective function and is globally optimal as well.

desired performance of the algorithm—the more samples generated, the more detailed the resulting sampled tree will be. The process goes as follows: alternately choose a random action and probabilistically take a transition according to $\vec{P}(t'_i|a,t)$. When a complete trajectory τ is reached, it is evaluated and optionally added to the set of sampled trajectories (denoted by $\mathcal{T}_S$). We will discuss this criterion further in Section 3.4.1.

We assume that it is most important to reduce error on the high-probability trajectories. This method focuses on reducing overall error in the approximation by ensuring that the areas of the trajectory space that could potentially contribute the largest local error are solved as accurately as possible. Even if the process "falls out of" the set of sampled trajectories during an episode (*e.g.*, if non-determinism causes an undesirable action outcome), we still maintain several nice characteristics. If our deviation from the sampled trajectory tree is near the root, it is likely that most or all of the subsequent trajectories have low probability in $P(\mathcal{T})$ (and were therefore not included in the set of sampled trajectories). On the other hand, if an evaluation sample falls out of the set of sampled trajectories far down the tree, it is likely that it will result in a trajectory with a high desired probability because that part of the trajectory space is more heavily represented in $\mathcal{T}_s$. Further, after reaching an unsampled part of the trajectory tree, an online "recovery" mechanism can be used. Earlier work [112] used a specialized search algorithm [135]. An alternate approach is to utilize an anytime sampling approach like the one we will present in Section 3.3.5 below. With or without a recovery technique, the effects of sampling error are minimized when it is possible to do well and are higher in the cases where it would have been impossible to do well. In Section 3.4 we will discuss and evaluate alternative approaches to sampling in much more detail.

3.3.5 An Online Algorithm

We now describe how the KL-divergence minimization can be applied in an online fashion. As formulated, we require $m(t)$ to be available for each local optimization. The requirement is actually weaker—all that is necessary is a function that, for a given prefix trajectory t, gives the *relative* masses required for each child of t. Specifically:

$$\forall t \to t', \;\; \tilde{m}(t') = c_t \cdot m(t') \;\; \text{where} \;\; c_t > 0 \tag{20}$$

This is because the local optimization in Equation 18 has the same maximizers even if $m(t')$ is multiplied by a positive scalar [16].

Below we derive an online algorithm when $m(t)$ can be computed quickly; however, we note that even when $m(t)$ cannot be computed efficiently, the offline sampling approach can be used. By processing the trajectories in the same order as before (*i.e.*, propagating upwards from the leaves) we are still able to calculate the $m(t)$ values as we need them, because of the following properties:

$$\begin{aligned} m(t) &= \sum_{t \preceq t'} m(t') & (21) \\ m(\tau) &= p(\tau) & (22) \end{aligned}$$

which follows from the definition of $m(t)$. In the event that the whole trajectory tree does not fit into memory, we can employ the same sampling technique used in the earlier approach.

To derive the online algorithm, one only needs to recall that we achieve the globally KL-optimal solution regardless of the order in which we perform the local optimizations (provided we have access to $m(t)$). One could start by processing the root of the trajectory tree, then process the children of the root, and so on, to compute a policy for each node in the tree. If $m(t)$ can be computed ahead of time (or efficiently in the background), then we can do even better by only solving the local optimizations that we absolutely must solve. This is done by interleaving the local optimization steps with taking actions in the world;

the local optimization tells us what the next action should be, and the action places us at a new node in the trajectory tree. Thus, we only solve the local optimization for trajectories we actually encounter. This is presented in Algorithm 2.

Algorithm 2 Online algorithm to minimize global KL error.

1: $t \leftarrow$ start state
2: **while** t is not a complete trajectory **do**
3: Compute the optimal local stochastic policy π_t^*:

$$\pi_t^* = \underset{\pi_t}{\operatorname{argmax}} \sum_{t \to t'} m(t') \log \sum_a P(t'|a,t) \cdot \pi_t(a)$$

4: Sample an action a from π_t^*.
5: Take action a in the environment, which will transition probabilistically to a new trajectory t' according to $P(t'|a,t)$.
6: $t \leftarrow t'$
7: **end while**

Note that we do not get a free lunch: without extra information about the nature of the trajectories or the analytic structure of $p(\tau)$, we are still limited by the complexity of the tree summation to compute $m(t)$, if we care to compute $m(t)$ exactly. This will be covered in more detail in Section 3.4.

On the other hand, requiring $m(t)$ to be of a certain easily computable form may be too restrictive. For instance, we could require that $p(\tau)$ be authored in such a way so that $m(t)$ can be computed solely from the prefix trajectory t, without having to do the tree summation. Unfortunately, this approach is unlikely to work in the general case where it is unreasonable to require that the quality of global outcome is uniquely determined by only local decisions.

This speaks to the need for online techniques to approximate $m(t)$. A direct and promising approach is to do *online sampling*, which we present in Algorithm 3. This is in direct contrast to the sampling techniques from the approaches presented above, which are performed entirely offline. In the online sampling approach, complete trajectories are sampled when $m(t)$ values are needed during the local optimization step. The maximum number of samples or the maximum size of the sampled tree can be fixed in order to be memory efficient. Furthermore, if the domain permits, the samples can even be collected in the background while other things are happening in the environment. The sampling would be paused whenever a local optimization step is reached, thus providing an estimate $\hat{m}(t)$ of $m(t)$ in an anytime fashion. As we move from one trajectory to a subsequent one, we can discard samples that no longer have an appropriate prefix, freeing up room for new samples. As we progress down the tree, the fixed number of samples are spread over fewer possible suffixes, thus providing better estimates of $m(t)$. As noted in early work on TTD-MDPs, the offline sampling approach performed well near the root of the tree and less well at the leaves [112]. This anytime approach will provide similar performance near the root and better, if not provably optimal, performance at the leaves.

Algorithm 3 Anytime sampling-based algorithm.

$t \leftarrow$ start state
$\hat{T} \leftarrow$ empty tree

Daemon Thread:
while t is not a complete trajectory **do**
  Generate a sample s such that $t \prec s$.
  Update the sampled tree $\hat{T}$ to include s.
  Update $\hat{m}(t)$ to reflect $\hat{T}$.
end while

Main Thread:
while t is not a complete trajectory **do**
  Compute the optimal local stochastic policy π_t^* as in Step 3 of Algorithm 2, using $\hat{m}(t)$ instead of $m(t)$.
  Sample an action a from π_t^*.
  Take action a in the world, which will transition probabilistically to a new trajectory t' according to $P(t'|a,t)$.
  $t \leftarrow t'$
  Prune $\hat{T}$ by removing samples s for which $t \not\prec s$.
end while

Note that we are leveraging the fact that $\hat{m}(t)$ only needs to provide a *relative* value (see Equation 20). The trajectory tree only needs to reflect a good sampling of the space, rather than provide a perfect estimate of $m(t)$, which would require sampling large portions of the tree. In fact, one needs only enough samples for the local optimization to be accurate, requiring far fewer samples than an approach that constructs an entire sampled tree to approximate the true (but possibly intractably large) tree. In addition, given enough time between decision points, the anytime approach will never reach an unsampled portion of the tree, a problem encountered by previous, offline approaches.

3.4 Authoring TTD-MDPs

As with any AI technique, a designer must specify each of the components. With TTD-MDPs, the number of valid trajectories is often large, so one cannot simply enumerate all possible trajectories and manually assign each one a probability weight. To be authorially feasible, there must be a compact way of specifying the target distribution.

In the original formulation of TTD-MDPs, an evaluation function that encapsulated the quality of a complete trajectory was used as the basis for the target distribution (much like the example in Figure 4). Insofar as that is a reasonable requirement, it is possible to use such an evaluation function to induce a distribution over trajectories. For instance, we may wish that trajectories occur with a probability proportional to their evaluation score: $p(\tau) \propto R(\tau)$. Unfortunately, such an approach still does not eliminate a difficult hurdle: solving a TTD-MDP efficiently and optimally requires $m(t)$ to be computed quickly. Below, we will describe two techniques that address this difficulty.

3.4.1 Sampling

The first authorial idiom we consider constructs an estimate for $p(\tau)$, from which we will compute $m(t)$. When an author has defined an evaluation function, we can use it to construct a distribution $p(\tau)$; however, doing so can be infeasible for large trajectory trees. Instead, we can approximate $p(\tau)$ by sampling a subset of trajectories $\mathcal{T}_s \subset \mathcal{T}$ via simulation of the process. We can then use $\widetilde{p}(\tau)$ as a replacement for $p(\tau)$, where $\widetilde{p}(\tau) \propto R(\tau)$

for $\tau \in \mathcal{T}_s$ and $\widetilde{p}(\tau) = 0$ otherwise. More precisely, we define:

$$\widetilde{p}(\tau) = \begin{cases} 0.0 & \text{if } \tau \notin \mathcal{T}_s \\ 0.0 & \text{if } R(\tau) < \phi \\ \frac{R(\tau)}{\mathcal{Z}} & \text{otherwise} \end{cases}$$

where ϕ is a threshold we select and $\mathcal{Z}$ is the normalizing constant. We construct $m(t)$ from this estimate; there will be an $m(t)$ value for each node in the trajectory tree induced by $\mathcal{T}_s$. Because we control the size of $\mathcal{T}_s$, we can adjust it to balance between our available memory and accuracy constraints.

There are several choices for generating samples. Following our early work [112], we could first select uniformly from the set of possible actions and then select uniformly from the set of successor trajectories, to generate a complete trajectory. An alternative is *Markov Chain Monte Carlo (MCMC)* sampling, a rejection sampling technique used to draw *i.i.d.* samples from a distribution that is difficult to sample directly. In our experiments, we use the Metropolis-Hastings algorithm [82, 46]. The pseudo-uniform sampling procedure described above can be used as the (unconditional) MCMC proposal distribution.

It is important to include the action in the sampling process as it constrains the set of states that can be reached. Consider actions a_1 and a_2, and partial trajectories t, t_1 and t_2, where t is parent of t_1 and t_2 in a trajectory tree. If $P(t_1|a_1, t) = 0.2$ and $P(t_2|a_1, t) = 0.8$, then both t_1 and t_2 are valid successor trajectories; however, if $P(t_1|a_2, t) = 0.0$ and $P(t_2|a_2, t) = 1.0$, then care must be taken because t_1 can never actually occur with action a_2. Further, in some domains reward is based on both the sequence of states and the actions taken by the system.

The sampling approach has its drawbacks. Due to the non-determinism in the environment ($P(t'|a, t)$) and the sheer size of the trajectory space, it is quite likely that an unsampled part of the full trajectory tree will be encountered during an episode. Presumably this is more likely in the low probability portions of the tree, so the process may have already been doing poorly to have entered into that part of the space. Further, if the deviation occurs near the leaves of the trees, it may be possible to perform online resampling to recover. In a domain studied in our earlier work [96], it appears that good trajectories often have common prefixes, so it may be that deviations are most likely to occur only after a good trajectory has already been ensured.

3.4.2 Prototypes

The second authorial idiom we consider computes $m(t)$ directly (which induces a target distribution $p(\tau)$ that is never represented explicitly) and is based on a set of prototypical "good" trajectories and a distance metric defined in trajectory space. Combining the distance metric with the prototypes can induce a probability distribution over all possible trajectories. One such method is to construct a Gaussian mixture model (GMM) over the set of prototypes:

$$m(t) = \sum_{i=1}^{N} p(\mu_i) \cdot \mathcal{N}(t; \mu_i, \sigma_i) \tag{23}$$

where

$$\mathcal{N}(t; \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}}\, e^{-d(t,\mu)^2/2\sigma^2}, \tag{24}$$

μ_i is a prototype and the centroid of a Gaussian distribution with variance σ_i^2, $p(\mu_i)$ is the prior weight given to each centroid which is a representation of the relative preference between prototype trajectories, and d is some distance measure on trajectories. There are a number of choices one could make for a distance metric. We explore three classes.

The first class is comprised of variations of *Levenshtein distance* or *edit distance* [62]. The edit distance is a computationally efficient generalization of the Hamming distance [44] that is defined over strings of unequal length and handles insertions, deletions, and substitutions. Consider three trajectories in our gridworld example: $t_1 = 1 \xrightarrow{R} 2 \xrightarrow{U} 5$, $t_2 = 1 \xrightarrow{U} 4 \xrightarrow{U} 5$, and $t_3 = 1 \xrightarrow{U} 2 \xrightarrow{U} 5$. The edit distance between t_1 and t_2 is $d_E(t_1, t_2) = 2$ because they differ in the first action and second state. By contrast, $d_E(t_1, t_3) = 1$.

There are several variations of edit distance. Let $l(t)$ be the length of a trajectory t

and $\rho(t,n)$ be the length n prefix of t; if $l(t) < n$, we define $\rho(t,n) = t$. Using $\rho(t,n)$, we can begin to construct measures of distance that are better suited to different domains. For example, in certain domains, deviations from desirable trajectories near the root of the trajectory tree are potentially more costly than deviations later. Thus, we may wish to consider a *scaled edit distance* between trajectories t and μ:

$$d_{SE}(t,\mu) = (1 + |l(t) - l(\mu)|) \cdot d_E(t, \rho(\mu, l(t)))$$

.

A second class of distance measures involves variations of the *longest common subsequence*. A subsequence of a trajectory is another trajectory formed by deleting some of the elements of the original trajectory without disturbing the relative position of the states (and actions). The longest common subsequence between two trajectories is the longest subsequence that appears in both strings.

A third class of distance measures uses an evaluation function directly when it is available. Typically, such functions are implemented as a linear combination of features about trajectories: $R(t) = \sum_k w_k \cdot f_k(t)$ (we refer the interested reader to [93, 135] for details). Here, distance from a prototype is simply defined as $d_F(t,\mu) = |R(t) - R(\mu)|$, which we shall call the *feature distance*. We could also construct a vector representation of these features $\vec{R}(t) = [w_1 \cdot f_1(t), w_2 \cdot f_2(t), \ldots]$ and use those vectors in a multivariate GMM. The weights on the features have an effect similar to changing the covariance matrix of the GMM, providing an interesting prospect for authorial control.

One problem with this approach is that features are not necessarily well defined over partial trajectories. We overcome this by defining a *blended feature distance* function:

$$d_{BF}(t,\mu) = \min\left[1, \frac{l(t)}{l(\mu)}\right] \cdot d_F(t, \rho(\mu, l(t))) + \max\left[0, \left(1 - \frac{l(t)}{l(\mu)}\right)\right] \cdot d_{\widetilde{E}}(t, \rho(\mu, l(t))) \tag{25}$$

where d_F is a function based on the features and $d_{\widetilde{E}}$ is some form of the edit distance. The first term on the right-hand side of Equation 25 represents increasing contributions of the drama management features as the length of trajectory t approaches that of μ. Similarly, the second term on the right-hand side of Equation 25 represents decreasing contributions from the edit distance as the lengths of the trajectories approach being equal.

Using prototypes provides a number of distinct advantages over sampling-based approaches. In comparison to authoring a reward function for an MDP, hand selecting a small number of prototypes may be significantly easier. Further, the prototype approach—especially using GMMs—allows efficient computation of $m(t)$ for partial trajectories. Even better, this approach provides a smooth distribution such that no trajectory has zero mass. Thus, it is not possible to fall out of the sampled space.

On the other hand, the problem of authoring has become the problem of choosing an appropriate distance function. When an evaluation function is available we can use it to capture subtleties in the values of states; however, when such functions are difficult to construct, it is unclear how well methods like edit distances will perform (it will be highly dependent on the details of the domain). Finally, prototypes must come from somewhere. They may be provided by the author, but they could also be generated by a sampling process like one of those presented in Section 3.4.1.

3.5 Concluding Thoughts on TTD-MDPs

Targeted Trajectory Distribution Markov Decision Processes are an MDP-based formalism that provide novel reasoning capabilities to agents. Most notably, since TTD-MDPs are defined over trajectories rather than states, they afford designers the ability to encode goals for agents that instruct them not just what to achieve, but how to achieve it. Further, the stochastic nature of the environment and target distribution provide a formalism for variety of experience—repeatedly applying the policy allows for the agents to achieve its goals in a number of different ways that are explicitly targeted and not solely influenced by the uncertainty in the dynamics. Lastly, with prototypes and the KL-divergence-based optimization method, an agent can solve a TTD-MDP online for the provably optimal solution. To date, this property has not been achieved in any other variants of the MDP formalism.

Because TTD-MDPs are not a relaxation of traditional MDPs, but a change in the notion of optimal, relaxations of the TTD-MDP model similar to some of those made of the MDP model can be used in the future to handle some of the increased complexity associated with real-world (or narrative) environments. For example, TTD-SMDPs could be studied to understand how agents can reason about probabilistic policies with non-atomic actions (see Chapter 5 for a derivation) or TTD-POMDPs could be studied to understand how agents can reason about trajectories through a space when the underlying world state is not fully observable.

In this chapter, we have presented the TTD-MDP formalism and illustrated how it is different from traditional MDPs. We have shown how this difference provides new reasoning capabilities for agents. We presented a variety of solution techniques with varying theoretical properties. Later in this dissertation, we will present the results of experiments that empirically illustrate the computational performance of these techniques.

CHAPTER IV

COMPUTATIONAL MODELS OF INFLUENCE FOR DRAMA MANAGEMENT

In this chapter we discuss our solution to the action/plan selection/generation and refinement problems for drama management (see Section 1.2). Our approaches use concepts from social psychology, discourse analysis, and natural language generation. Here we will motivate and present formalisms for both of these problems.

Interactive narrative experiences are marked by a strong social context. To leverage this social context, we developed computational models of *influence* and *persuasion* from social psychology and from *behavioral economics*. Our goals in developing computational models of influence were to: 1) provide authors with techniques designed to influence players to buy into the adoption of goals consistent with those the author has specified for the drama management system; 2) reduce the authoring complexity of scripted system responses to player actions in an interactive narrative by enabling authors to specify goals abstractly, relying on the principles of influence to bridge the gap to a concrete implementation in the virtual environment; and 3) accomplish (1) and (2) with the player feeling more engaged and without the player perceiving any decrease in self-agency.

Because these experiences are marked by a strong social context, relying solely on physical manipulation of an environment to engage in their management excludes a vast realm of possibilities. While there are a number of takes on the theories of influence and persuasion (an overview of which is well beyond the scope of this dissertation), we have based the work described in this dissertation on the theories of Cialdini [26] and Ariely [6]. We have chosen to base our work on these theories for a number of reasons including the respect their authors garner in their respective research communities. Additionally, and perhaps most significantly, these authors organize their research in a manner that we feel lends itself nicely to computational inquiry.

It is worth noting that these principles can ***never guarantee compliance***. We believe this is an important feature of our approach: the player always has the choice to disagree with what the DM is trying to get her to do. Thus, while the careful application of these principles of influence can greatly increase the chances of a player choosing to act in a manner consistent with the author-specified goals the system has, she always has a choice. Thus, the ***affordance for self-agency is strictly preserved***. The degree to which players perceive this will be discussed in Chapter 7.

In this chapter, we will provide a summary of influence concepts from social psychology. Building on those ideas, we will present a set of computational influence models that provide an abstract specification of how those tools of influence can be implemented. These specifications are representations of influence consistent with the second drama management question (action/plan selection/generation) described in Section 1.2. Lastly, we will provide some details on the concrete implementation of two of those models which provide a solution to the third drama management problem (action/plan refinement) described in Section 1.2.

It should be noted that the traditional approach of physical manipulation of the environment can place players in situations that may change their mental or emotional state. It is relatively straightforward to arrange for the transfer of knowledge from a non-player character (NPC) to the player; however, simply imparting knowledge to the player is not sufficient to increase the likelihood that she will choose to behave in a manner consistent with the system's goal for her (as specified by the author). Using influence theory, a drama manager will be able to decide how to change the player's mental or emotional state without using detailed pre-authored content.

The concepts discussed in the following two sections are excerpted and adapted from Cialdini's book *Influence: the Psychology of Persuasion* [26].

4.1 *Click Whirr*

All species—including humans—have certain built-in mechanical responses to specific stimuli. In animals, these responses take on many forms. For example, large species of fish often maintain a symbiotic relationship with another smaller fish known as a Bluestreak Cleaner Wrasse. The Wrasse eats parasites and dead tissue from the underside of the larger fish. The Wrasse will perform a dance in front of the larger fish which will activate its mechanical response and cause it to become perfectly still and wait to be cleaned. The Wrasse will then approach and clean it, obtaining an easy meal while providing a service to the larger fish.

These mechanical responses have been called "*click, whirr*" responses to represent the mechanical click of a recorded tape loading and the whirring of it as it is played. In animals, it is believed that these click whirr responses are instinctual and are free from social context. On the other hand, in humans it is believed that these responses are developed from psychological principles or social stereotypes that we learn over time. In fact, these learned responses in humans are thought to be coping mechanisms. We use them to reduce our cognitive burden when dealing with the ever-increasing complexity of stimuli we are faced with on a daily basis.

In order to use these principles effectively for interactive experiences, we need only to hit upon the trigger features that cause humans to play their recorded tapes. For example, a third species of fish, the Saber-Toothed Blenny, has learned to take advantage of the symbiotic relationship exhibited by the other two species simply by performing the dance to induce passiveness in the larger fish. When the larger fish enters its catatonic state, the Blenny will swim up and take a bite from the larger fish to obtain a free meal and swim away before it can be attacked. The amazing thing about using the principles of influence to the DM's advantage is that to do so requires minimal effort. As a result, a player willingly complying with the DM's (and therefore author's) wishes will tend to see their actions as a result of either their own choices or of natural forces rather than the influence of an exploiter [26].

4.2 *Tools of Influence*

In the long run, we plan to focus on six principles of influence. These principles have been identified by years of research in the field of social psychology and behavioral economics and are frequently employed as sales tactics by savvy marketers. There are other principles, but we have chosen to focus on these six in particular due to our belief that their organization is particularly amenable to operationalization for computational inquiry. These principles are:

- **Reciprocity**: give and take; when someone does something for us we feel obligated to return in kind (*cf.* [32, 101, 134])
- **Consistency**: we have an obsessive desire to appear consistent with what we have already done or said (*cf.* [19, 86, 98])
- **Social Proof**: we look to others, especially those similar to us, to determine the appropriate action to take (*cf.* [9, 10, 27])
- **Liking**: the more we like someone, the more willing we are to acquiesce to her requests (*cf.* [34, 35, 30])
- **Authority**: we have a deep sense of duty to authority (*cf.* [47, 84, 137])
- **Scarcity**: something that, on its own merits, holds little appeal to us will become decidedly more enticing if we perceive it will soon become unavailable to us (*cf.* [17, 40, 55, 83])

These principles provide the foundation for understanding how to create powerful tools to effect behavioral change in players. The models we are implementing are computational realizations of those tools. Used properly, each of these principles by themselves, or in combination with another, can greatly increase the likelihood of someone complying with a request.

Note that each principle can potentially be used in more than one way. For example, scarcity can be used to entice a player to obtain a particular object or to convince her that certain information she has obtained (or should obtain) is more important for success in the game. Liking can actually be employed as a function of friendship, reputation, or physical attractiveness. Authority can be asserted merely with a title like Dr. or can be a function of celebrity.

In addition to social psychology, many of these principles of influence are grounded in the theories of behavioral economics [6]. For example, the principle of social proof is often used in determining the appropriate price for a good. Similarly, consistency is referred to as *setting an anchor* and is discussed at length in the behavioral economics literature. Setting an anchor is something that humans do subconsciously when initially determining the value of something for which they were previously uncommitted. This could be an object like a TV or a time commitment like volunteering at an animal shelter. In a social context, setting an anchor refers to making a determination as to what some agreement is worth. For example, suppose we ask a player, "are you the type of person who is committed to protecting animals in need?" Many players, at least to appear caring, would answer yes to that question regardless of their level of commitment. Then, suppose we tried to determine exactly how committed the player is by asking, "given your commitment to protecting animals, how many hours per week of your time would you say it is worth?" At this point the player has already committed to wanting to help animals by answering yes to the first question, but by providing an amount of time when answering the second question, the player has set an anchor. Now, suppose the player has said yes to the first question and told us helping animals in need is worth three hours of her time per week. We now ask her, "will you spend an hour per week volunteering at an animal shelter?" Because she has already committed to helping animals and further anchored to three hours per week (even if only in theory), she is far more likely to agree to spend the hour per week at the shelter. The notion of setting an anchor in behavioral economics is a specialization of the commitment principle in social psychology. This particular example is further bolstered by the reciprocity principle as well. By reducing the time commitment after the anchor is set, the player is even more likely to concede.

4.3 Models of Influence

Given a goal selected by the DM (problem one from Chapter 1), in order to exert influence the system must perform two computations. First, it must pick a strategy for realizing the goal (problem two from Chapter 1). Traditionally this work has been performed by human designers specifying detailed instructions for the system. In our work, we provide influence as the framework that enables the system to reason about the strategy to use. In order to perform that reasoning, the system must have the appropriate schema available. We present ***influence acts*** as those schemata. For our purposes, we discuss the six principles described above (Section 4.2). These tools inform the design of schemata upon which higher level reasoning is performed. Additional schemata from other theories such as those described by Bauer and Levy [14], Goldstein, Martin, and Cialdini [42], Lakhani [57], or Thaler and Sunstein [124], to name a few, can be implemented at a later date.

As will be described in the next section, these tools of influence are encoded computationally in a schema-like language. Similar to discourse planning, actual speech acts must be generated from the schemata [48] (problem three from Chapter 1). Those speech acts can either be encoded by hand or generated by a natural language generation system. For our purposes, we chose to use a set of templates designed for use with the plan-like schemata. The templates provide some flexibility to the system. They are more general than hand-coded speech acts and more specific than a complete natural language generation system. In this dissertation, we use two sets of template models for two of the six tools of influence: reciprocity and scarcity (see Chapter 7 for more details).

We have opted to focus our work on two models of influence for three reasons. First, we feel it is necessary to conduct controlled—at least somewhat controlled—evaluations that try to limit the number of confounding factors. In order to fully evaluate all six influence models, the number of study participants and different treatment conditions would be prohibitively large at this stage of our research program. Second, our goal is to illustrate that it is feasible to use system-generated influence to shape player behavior rather than perform a complete investigation of numerous influence theories. And third, in order to effectively reason about which of the many influence tools to use, the system must have some representation of their effectiveness. Gathering data to construct accurate estimates is beyond the scope of this dissertation, although it is an important piece of future work on this topic. In Section 5.5 we will derive formulae for transferring effectiveness results from one domain (*i.e.*, the social psychology literature) to another domain (*i.e.*, a storytelling environment) to be used as an *a priori* estimate of effectiveness.

We have chosen to present example schemata for each of the six principles of persuasion to illustrate that their design and implementation is feasible. In order to facilitate a more tightly controlled study of their effectiveness, we opt only to evaluate two of those schemata: reciprocity and scarcity. Reciprocity and scarcity were selected because they illustrate both manners in which influence can be applied: through a call-and-response type interaction where the player must respond to an explicit question and without the need for such a response. Further, because hand-authored scarcity statements were used in the first study we conducted (Chapter 7), we felt it would be advantageous to evaluate the effectiveness of generated scarcity utterances as well.

4.3.1 Schemata Models

In the following sections, we will present each of Cialdini's six principles of influence in more detail. Additionally, for each of these principles we will present a detailed computational model. The models are somewhat abstract. They are specified in an AI planning-like

Table 1: A description of the parts of the schema models used to define influence schemata.

SCHEMA-NAME	
PARAMETERS: a list of parameters provided to the schema	
PRE: a description of the state of the environment before the schema can be applied	
ACTIONS:	
1:	a_1
·:	...
·:	A list of the actions taken in order to realize the postconditions.
·:	...
n:	a_n
POST: a description of the state of the environment after the schema has been applied	

structure inspired by work in discourse generation and understanding (*cf.* [58, 72, 85]).

We use a structure inspired by Rhetorical Structure Theory (RST) schemata to define our influence schemata [72]. RST schemata were developed as a tool to model discourse and ultimately led to a number of AI planning-based discourse generation systems (*e.g.*, [141]). RST leverages the insight that the structure of discourse is often more informative than the textual content. Through axiomatization a concise model can be developed to analyze and generate discourse [85]. In developing influence schemata, we leverage this same insight and process—the structure of an influence interaction is often more important than the specific communication and through axiomatization of influence we can develop a concise model to enable generation. While our schema model is inspired by the RST schema model, the structure itself has more in common with an ADL schema [41]. ADL is a language for specifying AI planning domains using operators and schema. An ADL operator is defined by three components: a name, preconditions, and effects. An ADL *schema* is an operator defined using variables as input.

Table 1 contains a definition of our influence schema structure. There are five main fields: name, parameters, preconditions, postconditions, and actions. The name, precondition, effects, and input variable fields are similar to ADL schemata. On the other hand, the actions fields are not represented in ADL operators. The actions fields of the schema contain a prescribed sequence of schemata or atomic actions that when applied in a state matching the preconditions will realize the effects. In practice, these actions are "templates" to be unified (see Section 4.3.2). It could be argued that the actions can be considered as recursively defined ADL operators. While functionally that representation would be equivalent to the influence schema structure we define, the ease of authoring would decrease. An action in the sequence of actions in an influence schema can be interpreted as syntactic sugar for an entire ADL operator that has the prior actions as preconditions. There is, however, one functional difference between our influence schema and a set of ADL operators. That difference is "jumps." In certain situations, it is necessary for a give-and-take to occur in order for influence to be exerted (*e.g.*, reciprocity via reciprocal concessions requires that the exploiter, or drama manager in our case, ask a series of questions to apply influence). To model the branching that occurs as a result of these give-and-take scenarios, the actions in our schema can be annotated with jump locations conditional on player responses (see Table 2 and Table 3 for two examples of jumps in our schema). This can be thought of as a form of contingency planning. This will be explained further in the next section.

The role of each of the fields in an influence schema will be illustrated in more detail in the sections below as example schemata for each of the six influence types are presented. To put these schemata to use in an interactive storytelling setting, the actions must be *refined* to a form that has concrete meaning in the environment. For the choose-your-own-adventure setting we use in this dissertation, the refinement process involves unifying these actions with natural language templates to produce text that is included in the story. This will be discussed in Section 4.3.2. In a more general storytelling setting, the refinement could involve physical manipulation of the environment such as adding or subtracting objects or having a NPC to interact with the player. We will briefly discuss these types of generalizations in the concluding chapter of this dissertation (Section 8.4.4).

4.3.1.1 Reciprocity

The notion of reciprocity is deeply engrained in us from the time we are young. The mantra "do unto others as you would have done to you" is a perfect example as well as the notion of "give and take." Here is an example. Two study participants (one real, one confederate) are rating paintings together, the confederate leaves the room to buy a Coke and brings one back for the actual participant. After the ratings are complete, the confederate asks the participant to buy some raffle tickets at a significantly higher price than the Coke. Participants bought raffle tickets twice as often when the confederate had given them a Coke [101].

Reciprocity often produces a "yes" to something that would otherwise outright be denied. To illustrate the power of reciprocity, the researchers conducting the above study examined how participants feelings toward the confederate affected response rates. To control for likeability, the participants in the study were asked about their feelings toward the confederate. When no Coke was provided (no favor performed), those that liked the confederate bought more raffle tickets; however, when the favor was performed by the confederate, the same response rate was achieved regardless of the participant's feelings toward the confederate.

Consider Hare Krishnas. They originally tried to fund raise by putting on an exhibition: exotic dress, dancing, and singing, which caused most people to not like them. Fund raising did not go well. They switched to the reciprocity rule by handing out flowers to passersby which did not require anyone to have any positive feelings toward them. They have been wildly successful. Over time, however, this tactic has been less successful—not because the rule of reciprocity has lost its magic, but because people have learned to recognize Krishnas and avoid them if at all possible. In other words, reciprocity is so powerful people would prefer to avoid a situation in which they are put to a decision using reciprocity.

This rule comes up in a number of places. In politics, congressmen doing favors for each other and lobbyists contribute to political campaigns. In the retail sector businesses use free samples that will get you exposed to a product (and hopefully like it), but is also a *free gift* that can evoke feelings of reciprocity. Restaurants in food courts at malls use this extensively. Similarly, car salesmen use this tactic when allowing perspective buyers to test drive vehicles. The salesmen have provided access to the product and spent time with you on the test drive, which can evoke a feeling of reciprocity.

A person can trigger a feeling of indebtedness by doing an uninvited favor for someone else. Another example of Cialdini [26] is that non-profits that send return address labels with their donation solicitations double their response rate. Interestingly enough, it is not necessary that a favor be requested in order to evoke the feeling of indebtedness. Going back to the Coke example from above, note that the only choices that are free of influence from the reciprocity rule are the choice made by the confederate: give a Coke to the other participant and then ask the other participant to purchase raffle tickets. The other participant, while in theory had free choice to reject the Coke or not buy tickets, was actually under the persuasive power of the reciprocity rule and did not have any choice that was free from a guilt-laden feeling. Interestingly, a small initial favor like giving someone a Coke can produce a sense of obligation to a substantially larger return favor like purchasing more expensive raffle tickets.

At a high level, the reciprocity rule goes as follows: a person who acts in a certain way toward us is entitled to a similar action in return. The first consequence of this rule is our propensity to repay favors. The second consequence of the rule is a propensity to make a concession to someone who concedes something to us. This variation of the rule indicates that people are inherently willing to compromise. This can be useful as a compliance technique by first making a large request we are sure will be denied and then retreating to a more reasonable, lesser, request that we actually want to have accepted. Note, this lesser request can be an objectively large one provided it is small in comparison to the first request. Unfortunately, if the initial large request is seen as too outlandish, this approach will backfire as any concessions after are not seen as actual concessions.

Table 2 contains a schema model for the reciprocity principle. The input to the model

Table 2: A schema model for the reciprocity principle that utilizes reciprocal concessions.

ASK-THEN-RETREAT
PARAMETERS: G, Cost(G) *PRE: Has-Goal(DM, G)*
ACTIONS: 1: *Knows-About(Player, G)* 2: *Ask-Commitment(Player, Goal(G), Inflate(Cost(G)):no,3:yes,0* 3: *Confirm-Rejection(Commitment(Player, Inflate(Cost(G))))* 4: *Ask-Commitment(Player, G, Cost(G))*
POST: Has-Influence(Player, G)

is the goal and cost to the player of achieving that goal. The cost can take on a number of forms including a time commitment, a monetary cost, or an opportunity cost. We will assume the input to the schema is valid but the preconditions must be satisfied separately (if they aren't already). They can either be realized by external actions that take place in the environment prior to the invocation of the schema, or actions can be taken to ensure the preconditions are satisfied before the schema's actions are executed. In this case, the only precondition is the drama manager having the input goal G, which will have to be true (otherwise the drama manager would not have invoked this schema with G as a parameter). Once the preconditions are satisfied, each of the four actions are executed in sequence, unless their output is already true in the environment as a result of some earlier or external actions. For example, the first action of the schema which ensures the player knows about the goal could be satisfied prior to execution. In that case, it would simply be skipped. Pay particular attention to the second action of this schema which ends with "*:no,3:yes,0*." This is the branching model mentioned above. If the player answers the question posed by action two with "no" then the third action is activated. On the other hand, if they answer "yes," then the schema exits successfully. If yes had been followed by -1, then the schema would have exited unsuccessfully. Once all actions in the schema have completed, the postcondition of the schema is set to true in the environment.

4.3.1.2 Commitment and Consistency

We, as humans, have a nearly obsessive desire to be (and appear) consistent with what we have already done. Further, this drive exists when it is something that we have said we are willing to do and have not necessarily done yet. Consider this example. A study was conducted on a beach with a staged theft. The researchers found that only four out of 20 onlookers made any attempt to stop the theft in general; however, if a nearby person was simply asked to "watch my stuff," the number who attempted to interrupt the theft jumped to 19 out of 20 and, in some cases, the thief was even physically restrained. The result: onlookers had a desire to be consistent with their commitment to "watch the stuff" [86].

In society, inconsistency is generally associated with indecisiveness, confusion, being two-faced, or being mentally ill. On the other hand, being consistent is generally associated with personal and intellectual strength. Further, it offers a shortcut through the density of modern life: once we have made up our mind on something, stubborn consistency enables us to not have to think hard about it again. Since this functions as a shield against thought, drama managers can use it to gain a compliance advantage.

By making a statement or choosing a side, the player is putting herself in a position to be stubbornly committed to it in the future regardless of whether or not it serves her interest. Commitment can be used to manipulate someone's self-image. Once the drama manager has obtained the desired self-image for the player, it can be used as an earlier commitment to coerce the player into doing anything that is consistent with that image. Here is another example from Cialdini [26]. Phone solicitations for donations are far more successful if the caller first asks "how are you?" and gets a reply of "good", "excellent", etc. The reason being that once someone has committed to being "good" they are in a position to help those that aren't.

We have an internal pressure to bring our self-image in line with our actions and an external pressure to bring our self-image in line with other people's expectations of us. Therefore, public commitments tend to be lasting commitments. In an experiment where three populations committed to an estimate publicly, privately, or mentally, it was found that they had varying willingness to change their estimate after evidence suggesting its incorrectness. Those that publicly committed were the least willing to change their estimate. Further, those that privately committed were far more likely to stick to their estimate than those that only mentally committed [26].

Social scientists have determined that we accept inner responsibility for a behavior when we believe we have made the choice absent of strong external motivation. Large prizes or serious threats are likely to produce immediate compliance but will not ensure the acceptance of inner responsibility. As a result, these techniques are unlikely to produce long-term commitment. Thus, the traditional "carrot and stick" methods used in computer games to get players to behave consistently with the goals the authors have specified may prove to be less effective in the long run than using psychological influence to get them to have an effect on their behavior.

As will be the case with most of the tools we discuss, the commitment and consistency principle can be leveraged for influence in a number of ways. For our purposes, we have chosen to highlight two of those ways: obtaining a verbal commitment from the player or leveraging a player's past action as an implicit commitment. Table 3 and Table 4 contain the two influence schemata for the commitment and consistency influence tool. In both of those schemata, the parameter H represents an action or series of actions. The idea underlying these schemata is to identify something related to the goal that the player will commit (or has committed) to doing to leverage influence.

4.3.1.3 Social Proof

The principle of social proof states that one means we use to determine what is correct is to find out what other people think is correct. It especially applies to determining what is correct behavior. For example, marketers love to inform us that a product is the "fastest growing" or "best selling" because they don't have to convince us that the product is good,

Table 3: The schema model for the commitment and consistency principle when a verbal commitment is used. Recall that the -1 in action 1 indicates the schema has exited unsuccessfully.

COMMIT-THEN-EXPLOIT
PARAMETERS: G, H
PRE: Has-Goal(DM, G) $\wedge$ *Consistent-With(G, H)*
ACTIONS:
1: *Ask-Commitment(Player, Idea-Of(H)):yes,2:no,-1*
2: *Confirm-Commitment(Player, Idea-Of(H))*
3: *Ask-Commitment(Player, G)*
POST: Has-Influence(Player, G)

Table 4: The schema model for the commitment and consistency principle when a player's past actions are used.

ACT-THEN-EXPLOIT
PARAMETERS: G, H
PRE: Has-Goal(DM, G) $\wedge$ *Consistent-With(G, H)* $\wedge$ *Has-Performed(H)*
ACTIONS:
1: *Confirm-Act(Player, H)*
2: *Ask-Commitment(Player, G)*
POST: Has-Influence(Player, G)

just that others have decided it is good.

Social proof can be so powerful that it can help us overcome fears. In one case, 67% of children who were afraid of dogs were convinced to play with a dog after just four days of watching another child play with the dog [9, 10]. In another study, children who were withdrawn from social situations quickly began to participate after watching a 23 minute video that showed withdrawn children joining group play. The effects of this influence was proven to be lasting as well. The times when we are unsure of ourselves or the situation is ambiguous are the times when we are most likely to look to and accept the actions of others as guidance for our own actions. Changing the actions of a few can have a significant impact on the actions of many others.

In addition to the condition of *uncertainty*, social proof works more effectively under the condition of *similarity*. That is, the principle of social proof works most powerfully when we observe the behavior of those most like us. Cialdini points out that marketers are

Table 5: The schema for the social proof tool when uncertainty about the situation is leveraged to create influence.

LEVERAGE-UNCERTAINTY
PARAMETERS: G, S, P, A
PRE: Has-Goal(DM, G) $\wedge$ *Situation(S)* $\wedge$ *In(Player, Situation(S))* $\wedge$ *Has-Uncertainty(Player, Situation(S))* $\wedge$ *Group(P)* $\wedge$ *Action(A)* $\wedge$ *Causes(A, G)*
ACTIONS:
1: *Tell(Player, Was(In(Situation(S), Group(P))))*
2: *Tell(Player, Performed(Group(P), Action(A), In(Situation(S)))*
3: *Ask(Player, Perform(Action(A))*
POST: Has-Influence(Player, G)

Table 6: The schema for the social proof tool when similarity to others is leveraged to create influence.

LEVERAGE-SIMILARITY
PARAMETERS: G, S, P, A
PRE: Has-Goal(DM, G) $\wedge$ *Situation(S)* $\wedge$ *In(Player, Situation(S))* $\wedge$ *Is-Similar(Player, Members-Of(Group(P)))* $\wedge$ *Action(A)* $\wedge$ *Causes(A, G)*
ACTIONS:
1: *Tell(Player, Is-Similar(Player, Members-Of(Group(P)))*
2: *Tell(Player, Was(In(Situation(S), Members-Of(Group(P)))))*
3: *Tell(Player, Performed(Members-Of(Group(P)), Action(A), In(Situation(S)))*
4: *Ask(Player, Perform(Action(A))*
POST: Has-Influence(Player, G)

increasingly using "testimonials" from "everyday people" about product quality because of the social proof theory [26]. Everyday people are more like us, so we are more likely to listen to their praise. We will use the actions of others to decide on behavior for ourselves, especially when we view the others as similar to ourselves.

Table 5 and Table 6 each contain a schema for the social proof influence tool. Table 5 describes how the player's uncertainty about how to act in a given situation can be leveraged to create influence. If the player's feelings about a situation are unknown (or can't be known), then the schema presented in Table 6 can be used. This schema leverages a (possibly hypothetical) group of other players who are similar to the player to create influence.

4.3.1.4 Liking

Unsurprisingly, we are more willing to accept the requests of friends than we are to accept the requests of complete strangers; however, strangers have many tools available to them that may make us just as willing to accept their requests. Luckily for strangers, there are many factors that can cause people to "like" others. One very important factor is *physical attractiveness* which seems to have is own influence response associated with it [35]. Further, attractiveness seems to enable people not just to influence the actions of others, but to influence the opinions of others more easily as well.

Another one of the most influential characteristics in terms of liking is *similarity*. We like people who are similar to us more than we like those that are obviously different. We are far more likely to comply with someone dressed in a similar fashion to us than someone who is not. This influence over liking goes beyond dress: political opinions, personality traits, lifestyle or background all appear to matter. Interestingly, there is already some evidence that gender-specific virtual characters can have a greater influence on the attitude change of humans of the opposite sex [144].

In the absence of other personal connections, *compliments* are the next dimension of liking used for social influence. Perhaps most interesting, studies suggest the compliments don't need to be accurate or relevant to have an influential effect [34].

Simple *association* can affect liking as well. It is clear that getting bad news from someone can cause us to like them less (hence the adage "don't shoot the messenger"). What is interesting is that the opposite is true as well. Consider a weather forecaster. People associate both the good and the bad weather with them and change their liking for him or her accordingly. As there is guilt by association, so too there is favor by association (think about good looking models in car advertisements). Isaac Asimov put it best when he said, "All things being equal, you root for your own sex, your own culture, your own locality...and what you want to prove is that *you* are better than the other person. Whomever you root for represents *you*; and when he wins, *you* win" [8].

Table 7: The schema for the liking tool when a compliment is used to gain the player's liking.

COMPLIMENT-TO-LIKE
PARAMETERS: G, A, C
PRE: Has-Goal(DM, G) ∧ Has-Performed(Player, Action(C)) ∧ Causes(A, G)
ACTIONS:
1: *Tell(Player, Compliment(Action(C)))*
2: *Ask(Player, Perform(Action(A))*
POST: Has-Influence(Player, G)

Table 8: The schema for the social proof tool when associating a positive is used to stimulate liking.

ASSOCIATE-TO-LIKE
PARAMETERS: G, A, E
PRE: Has-Goal(DM, G) ∧ Helps(Player, Entity(E)) ∧ Causes(A, G)
ACTIONS:
1: *Offer(Player, Entity(E))*
2: *Ask(Player, Perform(Action(A))*
POST: Has-Influence(Player, G)

Table 7 and Table 8 contain two schemata for the liking tool based on compliments and positive association. They have similar structures, but slightly differing preconditions and parameters. The COMPLIMENT-TO-LIKE schema (Table 7) is defined using an action C that the player has performed in the past; however, in general anything about the player can be used as the basis for a compliment (even if it doesn't apply directly). The association-based liking schema (Table 8) on the other hand is written more generally with an entity E. The entity could be something physical to give to the player or information that helps the player. Regardless of the form it takes, the same structure applies.

4.3.1.5 Authority

People have a deeply seated sense of duty to authority. In cases where an authority figure declares something or gives an order, what would otherwise make sense to us quickly becomes irrelevant. We react to the declaration or order rather than reasoning through the situation. This type of reaction occurs frequently in hospitals where doctor's orders are routinely carried out without question. In addition, the mere appearance of authority can sometimes be enough as in the case of a TV actor cast in the role of a doctor selling a product purportedly good for us.

We are often just as vulnerable to the appearance of authority as we are to the actual substance of authority. There are three main ways in which the appearance of authority can be created: titles, clothes, and trappings. Titles are both the hardest and easiest symbols of authority to obtain. To legitimately obtain a title, many years of hard work are required; however, it is easy to just assume a title and use it to your advantage (much as TV actors do). In one study, it was found that nurses were willing to administer a dose of an unapproved drug after an order came from an unknown physician over the phone [47]. The mere mention of the doctor title caused the nurse to respond automatically. Similarly, the things people wear can have an influence on our willingness to comply with them—especially if they wear a uniform. In many cases, people are far more willing to comply if someone is wearing official-looking attire. Further, business suits can have a similar effect as they appear to indicate the wearer is powerful and/or successful. Lastly, trappings are the final symbol of authority. Well-adorned clothes, expensive-looking accessories (*e.g.*, jewelry), and cars are all likely to indicate authority.

Because the appearance of authority is more often than not enough to increase compliance, it should be relatively straightforward to leverage authority for computational influence. Of the three varieties of authority-based influence, two are mainly visual in nature (uniforms and trappings) whereas the title variety can be communicated using language more easily. As such, we will only present a schema for that variety of authority-based influence in Table 9. Should an author desire to use an authority schema for a graphical interactive narrative experience, it would be a relatively straightforward exercise to write one for either uniforms or trappings.

Table 9: The schema for the authority tool when a title is used to inform the player of a person's authority and the person directly instructs the player to take an action.

LEVERAGE-TITLE	
PARAMETERS: G, A, P	
PRE: Has-Goal(DM, G) $\wedge$ Has-Title(Person(P)) $\wedge$ Causes(A, G)	
ACTIONS:	
1:	*Inform(Player, Title-Of(Person(P)))*
2:	*Suggest(Person(P), Player, Take-Action(Action(A)))*
3:	*Ask(Player, Perform(Action(A))*
POST: Has-Influence(Player, G)	

4.3.1.6 Scarcity

The influence principle of scarcity states: something that, on its own merits, holds little appeal to us will become decidedly more enticing if we perceive it will soon become unavailable to us. This is why people feel the need to answer the phone, especially when they don't know who is calling. By not answering the call, we may forever lose the information about who was calling and what they were calling about. In fact, the fear of losing something is far more motivating than the thought of gaining something of equal value. For example, it has been shown that pamphlets describing the importance of self breast exams for women are far more successful if they state what is to be lost by not performing the exams rather than what is to be gained by performing them [83].

One of the most straightforward uses of the scarcity principle is the "limited number" tactic. Potential customers are informed that there is a limited supply of a product that is likely not to be replenished. Similarly, it is often the case that salesmen will tell customers that a particular price will not be available if they don't purchase the product soon. This is the "deadline" tactic.

One feature of the scarcity principle is based on the theory of psychological reactance which states, albeit more eloquently, that we covet things more when we can't have them. One particularly interesting thing about this theory is we begin to see signs of it in children (as young as 24 months). This may indicate, unlike the other five principles of influence,

Table 10: The schema for the scarcity tool when limited supply is used to create the perception of scarcity.

REDUCE-NUMBER

PARAMETERS: G, O, Q, V
PRE: Has-Goal(DM, G) $\wedge$ *Has-Value(Object(O), Value(V))* $\wedge$
Has-Quantity(Object(O), Quantity(Q)) $\wedge$ *Associated-With(Object(O), G)*

ACTIONS:
1: *Inform(Player, Object(O))*
2: *Inform(Player, Has-Quantity(Object(O), Quantity(Q))*
3: *Inform(Player, Has-Value(Object(O), Value(V))*
4: *Inform(Player, Reduce(Quantity(Object(O))))*

POST: Has-Influence(Player, G)

that scarcity is less of a learned cultural response and more of an instinctual response.

There has been a study indicating overwhelmingly that banned material is more desirable to us and we therefore look upon it more favorably [17]. The most intriguing results of this study is that people tend to more willingly believe the message of what is banned without having heard it explicitly.

The idea that information that is scarce—that is information we believe we cannot obtain elsewhere—is more valuable tends to result in such information being more persuasive. In this light, Brock and Fromkin have developed the "commodity theory" of persuasion [40]. Cialdani's strongest evidence for this came in a study of beef sales where three populations of customers were: 1) given a normal sales presentation; 2) given the sales presentation with added information about an expected upcoming supply scarcity; and 3) same as (2) but also told the information concerning the scarcity was "exclusive." The results: group (2) bought twice as much beef as group (1) and group (3) bought six times as much beef as group (1) [55]. It turns out that the scarcity principle applies most when something that is known to be abundant suddenly becomes scarce.

Table 10 and Table 11 each contain a scarcity schema. Table 10 contains the schema for scarcity when limited supply is used as the trigger. This works only for objects, not for information or actions. When the objects are associated with a goal (a precondition of the schema), then scarcity can be used to influence the player toward the goal. In Table 11 there

Table 11: The schema for the scarcity tool when a deadline is used to create the perception of scarcity.

IMPOSE-DEADLINE
PARAMETERS: G, E
PRE: Has-Goal(DM, G) $\wedge$ Associated-With(Entity(E), G)
ACTIONS:
1: *Inform(Player, Entity(E))*
2: *Inform(Player, Last-Opportunity(Exercise(Entity(E))))*
POST: Has-Influence(Player, G)

is a schema for scarcity when a deadline is imposed. This schema uses an "entity" rather than an object because it can be applied to either objects or actions. The statement "*Exercise(Entity(E))*" in action 2 of the schema will ground out to either "Obtain(Object(E))" or "Perform(Action(E))" depending on the type of E.

4.3.2 Template Models

The schemata presented in Section 4.3.1 provide the foundation upon which discourse-style plans can be generated. Although we will not discuss nor evaluate a planning algorithm that reasons using these schemata, they are important for illustrating that such an algorithm could be implemented. Further, an algorithm that uses influence schemata would provide a computational solution to the second design problem discussed in Chapter 1: Given the sequence of plot events that have occurred thus far and the next plot event goal, the system must select or generate a plan of action (generally not atomic) to achieve that goal.

The selection of one or more influence schemata to create a plan for achieving a plot event goal does not provide the ultimate solution. Comprised of generic predicates rather than atomic utterances, the schemata are still relatively abstract. Therefore, to bridge the gap between selecting a plot event goal (problem one from Chapter 1) and the concrete game environment, another step is needed. To accomplish this, we turn to a set of pre-authored templates. Templates are often used in natural language generation systems (*cf.* [20, 125, 131, 136]) and have been argued to have equal quality to "real" natural language generation systems in terms of maintainability, linguistic well-foundedness, and quality of

Listing 4.1: An example of an ask-commitment template.

```
name: ask-commitment
type: sentence
num_params: 2
param_name_1: <player>
param_name_2: <phrase>
param_type_1: noun_proper
param_type_2: verb_clause
text: Is it worth it for <player> to <phrase>
punctuation: ?
```

output [130].

For each influence schema, there are numerous ground realizations in the narrative environment. The process of "compiling" a schema to atomic actions is known as "***refinement***". The method we have chosen for refinement is to perform template unification. For each schema, we can have a set of templates that distinguish between some of the subtleties associated with the parameters such as singular vs. plural or object vs. action. Additionally, there can be more than one template for a schema that can be applied given the current parameters. Randomly or heuristically choosing a template from those that are applicable will allow for variety in the refinement process. This will hopefully prevent excessive repetitiveness for players.

van Deemter, Krahmer, and Theune describe a template as a "linguistic structure that may contain gaps" [130]. They claim that well-formed linguistic structures are realized when the gaps are filled, or replaced with linguistic structures that contain no gaps. Canned text is therefore a boundary case of a template without gaps. The use of canned text would limit the applicability of our models. The use of gaps, or variable bindings, in our templates allows for a reasonably-sized set of templates to be used in a variety of situations during the story.

In our system, templates are characterized by a number of fields: name, type, number of parameters, parameter names and types, the text to be produced, and optionally punctuation. Listing 4.1 contains an example of an "ask-commitment" template for our system that has type "sentence" and takes two parameters. This template would be applicable when the system intends to obtain a commitment from the player, perhaps for the reciprocity or commitment tools. We implement a number of such templates all designed to fulfill the predicate *Ask-Commitment(Player, Idea-Of(H)):yes,2:no,-1* (action 1 in the schema from Table 3). Thus, by providing a number of alternatives for each of the actions in a schema, the system can dynamically create sequences of text specifically tailored to the situation facing the drama manager. To do so, the system selects the template appropriate for the parameter variables provided to the schema and passed as input to the unification procedure. A heuristic can be used to disambiguate among multiple matched templates and the variables are then bound to the "gaps" in the templates.

4.4 Concluding Thoughts on Influence Schemata

In this chapter we have summarized some results of the social psychology literature on applying influence to induce compliance. Along with that summary, we have operationalized those concepts using concise schemta representations that form the basis upon which the action/plan selection/generation problem can be solved (problem one form Section 1.2). For each of the six types of influence discussed, we have provided at least one example schema to illustrate in some detail how influence can be operationalized. We have also discussed the use of natural language templates for the action/plan refinement problem (problem two from Section 1.2) in our evaluation environment. In Chapter 7 we will report on the effectiveness of using two types of influence schemata, scarcity and reciprocity, for guiding players through our choose-your-own-adventure storytelling environment.

CHAPTER V

RECONCILING TTD-MDPS AND INFLUENCE SCHEMATA

This chapter is devoted to some glue that connects two of the solutions (presented in the previous two chapters) to two of the drama management problems (presented in Section 1.2). Here we derive extensions to the formalism and algorithms designed for the goal selection problem that will enable integration with the formalism designed for the action/plan selection/generation problem.

We discuss extensions to the TTD-MDP formalism necessitated by our computational influence models. As with most work on traditional Markov Decision Processes, the discussion of TTD-MDPs presented in Chapter 3 is based on an assumption of atomic actions. Atomic actions are assumed to occur in isolation. They take unit time, cannot be preempted, nor can they be interrupted. Once an action is taken, it runs to completion without interference from outside forces.

Creating influence, however, is not atomic. Even at its simplest, the exertion of influence occurs after a careful sequence of actions have been taken. In the general case each one of those actions may independently fail or the entire composite sequence of actions, even if successfully completed, may fail to create influence. By considering the effects of non-atomic actions, we can adapt the TTD-MDP model appropriately so the conditions that enable efficient computation are not violated.

In order to make these adjustments to the TTD-MDP model, we consider work on "Semi-Markov Decision Processes" (SMDPs) as inspiration. In SMDPs, the assumption of atomic actions made in MDPs is relaxed. Similarly, in this chapter we describe a simplified TTD-SMDP model where the assumption of atomic actions made in TTD-MDPs is relaxed. We note that this is not a fully generalized TTD-SMDP model; it is designed specifically with the application of influence models in mind.[1] In Section 5.2 and Section 5.4 we will discuss the simplifying assumptions we make. The model we present is based on the theory of options in reinforcement learning [123].

5.1 The Options Framework for SMDPs

In reinforcement learning terms, an ***option*** is a closed-loop policy for taking an action over a period of time (*nb.* that period of time can be atomic, or longer) [123]. They are a tool for including temporally extended knowledge in the reinforcement learning framework. Sutton, Precup, and Singh have shown that options can be used interchangeably with actions in planning and learning methods. A similar result will prove true for options and TTD-MDPs.

More formally, an option in an MDP is defined by three components: a ***policy***, a ***termination condition***, and an ***initiation set***. Therefore, the formal specification of an option is a tuple $< \pi, \beta, \mathcal{I} >$. Here is what each of the components of the option provide:

- The policy $\pi : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$ specifies a probability distribution over action choices in the given state. This specification may be deterministic. A special case of an option is one where there is only one atomic action such that for all states in the initiation set the policy evaluates to exactly 1.0 and evaluates to 0.0 for all other atomic actions and the states reached using that action satisfy the termination condition with probability 1.0

- The termination condition $\beta : \mathcal{S} \rightarrow [0, 1]$ is a function that gives the probability the option's policy will terminate if the given state is encountered. During application of the policy, the system evaluates the termination condition using the current state as the input to determine if another action should be taken according to the policy, if another option should be selected, or if the option should terminate

[1] It would not be difficult to specify a fully generalized TTD-SMDP model; however, we do not require the representational power of such a model to accomodate influence schemata for drama management using TTD-MDPs. Thus, we opt for a simplified TTD-SMDP model to reduce complexity and improve readability.

- The initiation set $\mathcal{I} \subseteq \mathcal{S}$ is the subset of states in which the option can be applied

Note that this description is of a ***Markov option***. There are also ***Semi-Markov options*** in which the policy and termination conditions are defined over histories of states Ω as well: $\pi : \Omega \times \mathcal{A} \rightarrow [0, 1]$ and $\beta : \Omega \rightarrow [0, 1]$. Semi-Markov options are the types of options that model influence schemata as processes in which the history of actions taken is what determines if influence is exerted or not—influence is not atomic. Using this framework, we can consider the decisions made by drama managers to be among options, some of which only last for a single time step (traditional atomic actions), others of which are temporally extended. Regardless of duration, all options can be treated the same way.

To explain options more intuitively, consider the task of a robot moving an object from one position to another. There are a few levels at which this problem can be reasoned about. For example, the robot could reason at a relatively abstract level using action-types such as MOVE-TO, PICK-UP, PUT-DOWN, etc; however, in order for these actions to take place in the environment, there are a number of lower-level behaviors that must be accomplished. In order to PICK-UP an object, the robot must be within reach of the object and know the position of the object relative to its own position. Then, it must determine the appropriate speed with which to maneuver its hand into position to grasp the object, determine the appropriate pressure to apply to grasp the object, etc. The robot must know when the object is in its grasp and accordingly exit the PICK-UP option.

The relatively abstract reasoning the robot performs is in fact reasoning over options—not actions. The PICK-UP action in practice is implemented as a sequence of lower-level actions, which in turn may also be a sequence of even lower-level actions, *etc.* In order to execute the higher-level actions, the robot implements a plan and, if necessary, performs replanning to handle unanticipated changes in the environment.

5.2 *The Options Framework for TTD-SMDPs*

Using non-atomic influence actions does not change any of the theoretical characteristics of the TTD-MDP model presented in Chapter 3. Here, and in the next section, we will present this more precisely. As the difference between an SMDP and an MDP is non-atomic options instead of atomic actions, so too is the difference between a TTD-SMDP and a TTD-MDP the use of non-atomic options in place of atomic actions. The four main components of a TTD-MDP specification are the same for a TTD-SMDP specification. Therefore, the presentation in this section is focused on the definition of options for TTD-SMDPs.

First, we define some new notation. Let o be an option. Here, we will use o in place of the actions a that were used in Chapter 3. Additionally, we will refer to the states underlying a TTD-SMDP here as *events* e. We denote events e to differentiate them from *internal states* u which are irrelevant to the definition of the TTD-SMDP. Let $\mathcal{U}$ be the set of internal states. A *trajectory* in a TTD-SMDP, for the purposes of this discussion, will consist of an alternating sequence of events e and options o.

For TTD-SMDPs, as for SMDPs, an option is defined by a tuple $< \pi, \beta, \mathcal{I} >$. Here is what each of the components of the option provide for TTD-SMDPs:

- The policy $\pi : \mathcal{U} \times \mathcal{A} \rightarrow [0, 1]$ is a specification of the probability that the system will perform an action in a particular *internal state*. This specification may be deterministic. An internal state differs from the notion of an event that underlies a trajectory in that it is insignificant to the trajectory itself. Consider the robot example from the previous section. There, the events might include one where the robot possesses an object and one where it does not. The internal states will represent the robot's proximity to the object, the position of it's actuators, etc. From the perspective of constructing a trajectory, the proximity of the robot to the object is irrelevant—what is important is whether or not the robot possesses the object (the events). A special case of an option is the atomic action where the policy only contains one action applied in the states of the initiation set and the states reached using that action satisfy the termination condition

- The termination condition $\beta : \mathcal{S} \cup \mathcal{U} \rightarrow [0, 1]$ is a function that gives the probability the option's policy will terminate if the given event or internal state is encountered. During application of the policy, the system evaluates the termination condition using the current state as the input to determine if another action should be taken according to the policy, if another option should be selected, or if the option should terminate

- The initiation set $\mathcal{I} \subseteq \mathcal{S}$ is the subset of *events* in which the option can be applied

Consider the following trajectory t that consists of a sequence of n events and options:

$$t = e_0 : o_0 : e_1 : o_1 : \ldots : e_n$$

Note that t terminates in an event e_n. Trajectories that can occur immediately subsequent to t are the set $\{t_i | t_i = t : o : e_{n+1}\}$. There are a few constraints on each t_i. First, the final event of t (*i.e.*, e_n) must be in the initiation set of o, and second, e_{n+1} must have $\beta(e_{n+1}) = 1.0$ in the termination condition of o. We require $\beta(e_{n+1}) = 1.0$, rather than the more general $\beta(e_{n+1}) > 0.0$, because in the storytelling context we want the DM to begin reasoning about how to shape the player's next decision once a new plot event occurs. If we allowed $\beta(e_{n+1}) > 0.0$ there could be some probability that the DM would continue deliberations on shaping a decision the player has already made.

Each option is defined by the components described in Section 5.1. Thus, for option o taken in trajectory $t = e_0 : \ldots : e_n$ we have

$$o = < \pi, \beta, \mathcal{I} > \tag{26}$$

$$= < \pi : \{u_{i,j}\} \times \{a_{i,j}\} \rightarrow [0, 1], \beta : \{u_{i,j}, e_{n+1,1}, e_{n+1,2}, \ldots\} \rightarrow [0, 1], \{e_n\} > \tag{27}$$

where $\{u_{i,j}\}$ is the set of internal states that occur between event e_n and possible subsequent events $e_{n+1,i}$. The policy π maps internal states $u_{i,j}$ to atomic actions $a_{i,j}$. Note that in the general case these atomic actions could be recursively defined as options. The notation we use to describe actions can be thought to define equivalence classes among action types. The two subscripts on actions, i and j, are indices indicating the order in which the action classes are selected by the policy (i) and indicating the number of times an action of that class has been instantiated (j). An alternative interpretation of the action subscripts is that i indicates the action class and j indicates the particular action within that class. The two subscripts on states, i and j, are indices indicating the action class after which this state type occurs (i) and indicating the variant of that state type (j). Thus, a state $u_{4,2}$ would occur after an action class $a_{4,i}$ had been executed and the process resulted in the second variant of the state type.

Recall that actions are either atomic or recursively defined as options and are grouped into equivalence classes based on the internal state they are trying to realize. Since actions can fail to realize the internal state they are targeting, we allow for the possibility of re-execution at a later time. In a fully general model of TTD-SMDPs, we would allow for "out of order" re-execution of an action from an action class that has previously failed. In other words, the following action sequence might be possible: $a_{1,1}, a_{2,1}, a_{1,2}$; however, in the simplified TTD-SMDP model for DODM and influence schemata, we do not need to allow that behavior and therefore require that all instantiations of an action from class i occur before any instantiation of an action from a class k where $i < k$. See Section 5.4 for an explanation of why.

Repeated action class failures are handled in our model by requiring that the policy jump to a "failure termination state" once a certain number of action failures have been reached for the same action class. For example, suppose actions in class i were intended to realize internal state $u_{i,1}$ and the failure limit is set to three. Then, the following sequence would result in the option failing: $u_{i-1,j} : a_{i,1} : u_{i,2} : a_{i,2} : u_{i,2} : a_{i,3} : u_{i,2}$. Here the policy would terminate and the option would exit. On the other hand, the following sequence would be valid: $u_{i-1,j} : a_{i,1} : u_{i,2} : a_{i,2} : u_{i,2} : a_{i,3} : u_{i,1} : a_{i+1,1}$.

The termination condition is met in one of three ways: 1) one of the subsequent events $e_{n+1,i}$ is reached before the policy completes; 2) a long enough sequence of repeated action class failures occurs; and 3) the policy completes successfully (potentially with some action failures, but never more than the threshold in sequence). In any of those three conditions, $\beta = 1.0$ and in any other condition, $\beta = 0.0$.

5.3 Tracing an Example

Now that we have presented a formal definition of our DODM/influence-appropriate simplified TTD-SMDP model, let us take a step back and trace through an example. Given a particular partial trajectory, the drama manager must select from the possible ***options*** in order to take action to realize the desired subsequent trajectory. In Section 5.4 we will describe how the probabilities traditionally associated with actions in the TTD-MDP transition model can be associated with options in the TTD-SMDP formalism. For the purposes of this discussion, it is safe to simply assume they exist and the drama manager has access to them.

Only options for which the current trajectory/event are in the initiation set will be considered, so any option the drama manager selects will be applicable. Further, all options will assign all of the events directly reachable from the current partial trajectory probability 1.0 in the option's termination condition and all other events probability 0.0. This ensures the option terminates when the player acts to advance the plot.

These conditions ensure that the option selected by the drama manager will be applicable in the current story state. Execution of the option proceeds by selecting and executing lower-level atomic actions (or recursively executing another option), observing the "internal state" that is reached, and repeating until the termination condition is met. We note the atomic actions or recursive options by $a_{i,j}$ (or $a(i, j)$ in Figure 6) and internal states by $u_{i,j}$ (or $u(i, j)$ in Figure 6).

Figure 6 is a graphical representation of how an option execution might occur. In that figure, we have depicted a scenario where the option policy specifies three actions to be taken in sequence, and after each action there are two possible resulting states $u_{i,1}$ (success) and $u_{i,2}$ (failure). Each action can be taken at most three times $a_{i,1}, a_{i,2},$ and $a_{i,3}$ before the option is considered to have failed. In the context of influence, this would be akin to replanning if a discourse plan fails, but choosing to give up after a fixed number of failed attempts.

In Figure 6, there are two ways in which the option policy terminates: 1) it "fails" three consecutive action executions of the same type (*i.e.*, enters internal state $u_{i,2}$ three times); and 2) it reaches state $u_{3,1}$ without failing any action class three consecutive times. In Figure 6 we have represented the successful and unsuccessful completion of the action sequences by including two absorbing states. Strictly speaking, these absorbing states are an alias for the $u_{i,j}$ internal states that precede them, indicating the equivalence class of internal states resulting from either successful on unsuccessful completion of the policy. From these two alias states, one of three subsequent story events $e_1, e_2,$ or e_3 comes next (which all have probability 1.0 from the option's termination condition).

Figure 6 is included both to make precise what is written in text, but also to illustrate a point: even in a small example with three action classes, two internal states per action class, and a limit of three consecutive action class failures before termination, the option space is quite large. Therefore, for the purposes of illustration we have included Figure 7 which depicts a further simplified example. The same mechanics apply, but the example in Figure 7 contains only one action class, two internal states per action class, and a limit of two consecutive action class failures before termination.

In order for the TTD-MDP calculations presented in Chapter 3 to be performed, certain probabilities must be available to the system. Specifically, the transition probability that the subsequent trajectory will be $t : o : e_i$ given that the option o is applied in trajectory t, denoted $P(t : o : e_i | o, t)$, must be available for each subsequent story event e_i that directly succeeds t. For notational convenience, we may leave out the first o in the transition model

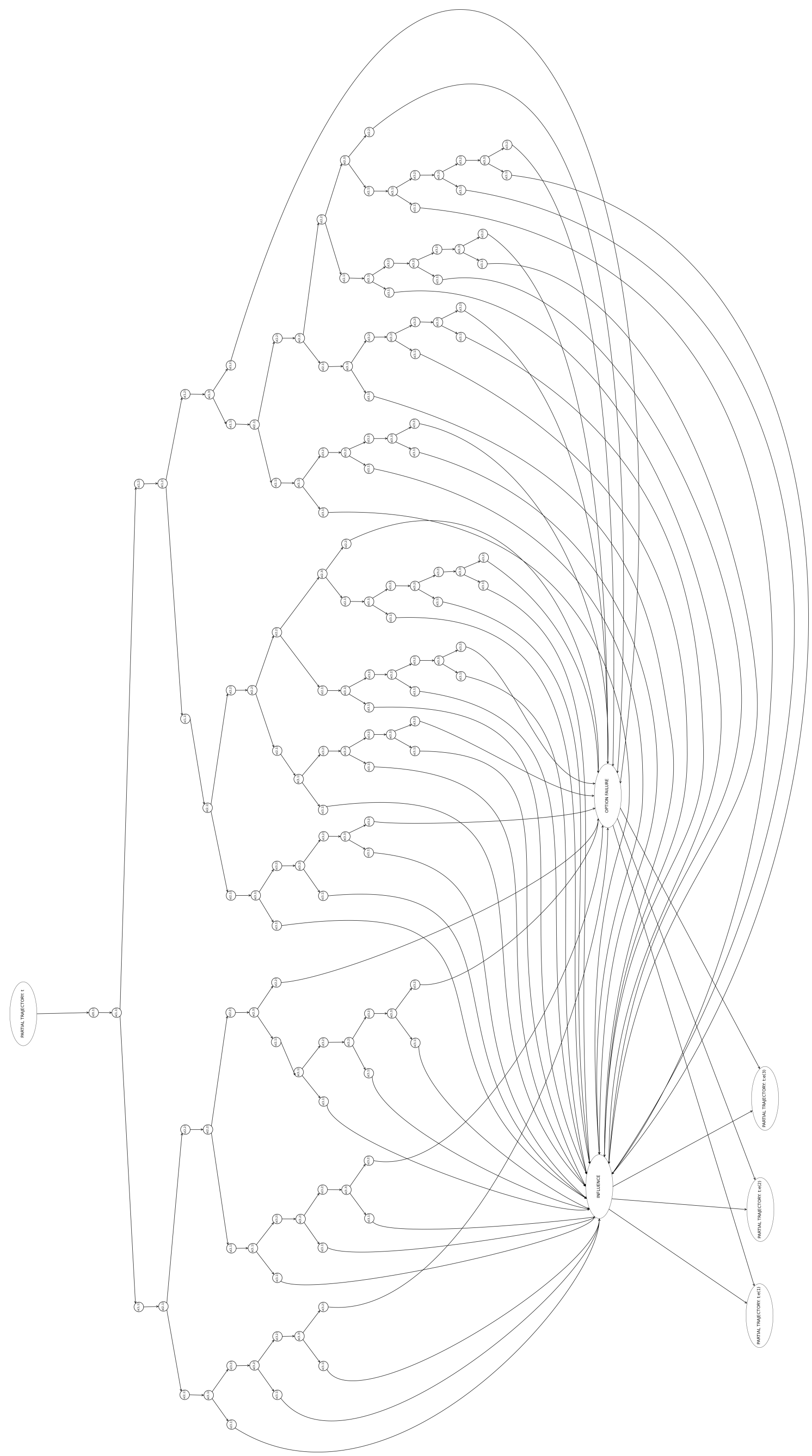


Figure 6: A trace of a simple option with three action classes, two internal states per action class, and a limit of three consecutive action class failures. Note the landscape orientation. Note that we have included this figure even though it is difficult to read to illustrate the point that even a simple TTD-SMDP option example induces a significantly large option space.

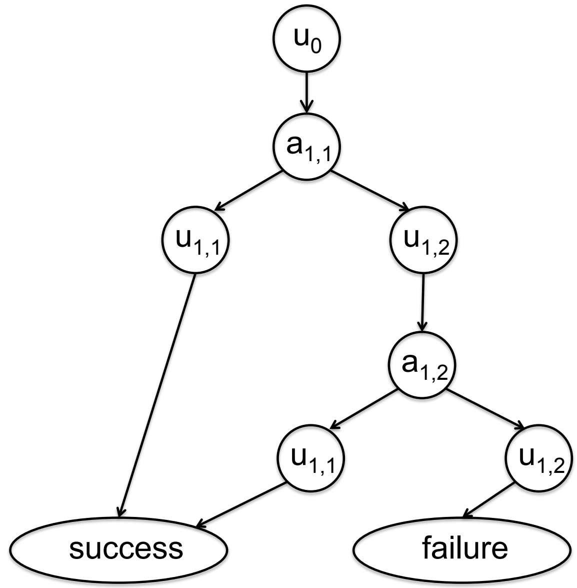


Figure 7: A trace of a simplified example with one action class, two internal states per action class, and a limit of two consecutive action class failures.

and write it as $P(t : e_i|o, t)$. It is possible to use the structure presented in Figure 6 to inform the computation of those probabilities from lower-level action/option executions. We will present this computation in Section 5.4 below.

5.4 Drama Management Interpretation and Simplifying Assumptions

In this section, we will relate the pieces of the TTD-SMDP model presented above to the drama management concepts discussed in Chapter 1 and influence concepts discussed in Chapter 4. First, we will discuss the relationship of the full model to the reasoning and action implementation a drama manager uses. We will then discuss how the transition probabilities necessary for the solution of a TTD-MDP as presented in Chapter 3 can be computed from the TTD-SMDP model. Next we will present a number of simplifying assumptions that make this process even easier in the evaluation environment we will present in Chapter 7. Lastly, we will derive a set of formulae that prescribe a method for transferring transition probabilities between domains in a meaningful way.

5.4.1 TTD-SMDPs and Drama Management

In the DODM model of drama management, the drama manager reasons about abstract plot events and selects abstract actions. What occurs between these abstract plot events is—given the right plot abstraction—inconsequential to the reasoning process, especially when simulation is used to obtain results.

The option components (π, β, and $\mathcal{I}$) each relate to the specification of a DODM drama manager. In particular, the initiation set for the option $\mathcal{I} \subseteq \mathcal{S}$ is comprised of the terminated story events of partial trajectories from the TTD-MDP specification. The termination condition is specified over internal states as well as DODM plot events. This ensures that if the player takes action before the option completes execution, the option will terminate properly. Lastly, the policy π can be defined by the influence schemata discussed in Chapter 4.

In the TTD-SMDP model, the four components that specify the drama management problem are essentially the same as the TTD-MDP components: 1) a set of plot events with precedence constraints; 2) a set of drama manager ***options***; 3) a player model specifying transition dynamics between plot event sequences and subsequent plot events ***conditioned on drama manager options***; and 4) a target distribution indicating relative preference for trajectories. The two highlighted portions of this list are the subtle differences between the TTD-MDP and TTD-SMDP model. The first, a "set of drama manager options," is described at length in the preceding sections. The second, transition dynamics "conditioned on drama manager options," is the topic of this section.

Previously, transition dynamics (given by $P(t'|a, t)$) were specified at the plot event and action level. With the introduction of non-atomic options, these probabilities now need to capture all of the possible paths the through the option space and its internal states that lead to the next plot event: $P(t : e_n|o, t)$. In the general case, this can be a large and potentially complex calculation. The example discussed in Section 5.3 and shown in Figure 6 is a relatively small one. Even in this small example, the value of $P(t'|a, t)$ is computed from the probabilities of 45 individual paths. More precisely, the probability $P(t : e_n|o, t)$ of reaching an event e_n given option o has been selected, and the process is in trajectory t is given by the sum of the probabilities of all paths starting in $t : u_0$ and ending in e_n, which in turn are given by the product of the sequence of action choices and internal state outcomes.

Here, let $\sigma = \sigma_0^u : \sigma_1^a : \sigma_1^u : \ldots : \sigma_n^a : \sigma_n^u : e_n = u_0 : \ldots : a_{i,j} : u_{i,k} : \ldots : e_n$ be a path induced by the application of option o. σ_m^a represents the action $a_{i,j}$ taken at the m-th action-state iteration of the option application and σ_m^u is the resulting state $u_{i,k}$. Thus, we have the probability of e_n given the path σ defined by:

$$P(t : e_n|s, t) = \prod_{i=1}^{n} P(\sigma_i^a|\sigma_{i-1}^u) \cdot P(\sigma_i^u|\sigma_i^a) \tag{28}$$

This represents the general case. In fact, this computation is slightly easier in our case as we are assuming that the option policy π is deterministic. Thus, each action choice contributes a probability $P(\sigma_i^a|\sigma_{i-1}^u) = 1.0$ and can safely be excluded from the computation. Further, unlike in the top-level TTD-MDP computation, complete option histories are not necessary. As a result of the "stopping condition" encoded in the policies and described in Section 5.2, σ_i^a must be unique for all i. Therefore, the proper probability mass is contributed to the computation without the need for history to indicate uniqueness.

This, however, is only part of the picture. This gives the probability of reaching e_n on any one path. To get a probability value $P(t : e_n|o, t)$ that represents all ways of reaching $t : e_n$, all possible paths of action-internal-state sequences must be considered. Thus, let $\Sigma_{e_n} = \{\sigma|\sigma$ is a path from u_0 to e_n obtained by applying $o\}$. Then,

$$P(t : e_n|o, t) = \sum_{\sigma \in \Sigma_{e_n}} P(t : e_n|\sigma, t) \tag{29}$$

Given this value, the TTD-MDP computation can proceed as described in Chapter 3.

5.4.2 Simplifications Induced by Influence Schemata

Because we know that $P(\sigma_i^a|\sigma_{i-1}^u)$ is defined by the deterministic policy, the question remains as to where $P(\sigma_i^u|\sigma_i^a)$ and where $P(e_n|s_n^u)$ come from. In general, these values will need to either be estimated by an author *a priori* or obtained empirically. All of the existing work on DODM has relied on authored estimates of player behavior—a knowledge engineering approach. Ultimately, it is our goal to run extensive enough tests with players where we can obtain an accurate *a priori* probability estimate; however, doing so will require extensive experimentation and data collection that is well beyond the scope of this dissertation. See Section 5.5 for a strategy to obtain the *a priori* estimates from data reported on other domains.

Luckily, we have an advantage. In Chapter 7 we will present a web-based choose-your-own-adventure-style storytelling system that forms the platform for evaluating our approach with user studies. The system works by iteratively presenting text, optionally videos, and a question to the player. The player's answer to the question determines the subsequent story event. In this setting, the drama manager options are implemented by generating additional text (using the influence schemata) to present to the player. As a result, failures both at the option level and at the lower action level do not occur. When the system implements an influence schema as an option, it triggers the execution of each of the actions in the schema. These actions trigger the unification of a natural language template with the current state of the system. While in theory this unification process could fail, we can engineer our template sets so that, in practice, for the domain we are using for testing, such failures do not occur.

To understand why, let us relate the actions and internal states of the option specification to the influence schemata. First, the actions specified in the schema definition as well as the preconditions on the schema inputs comprise the set of actions in the option policy. Depending on the current state of the story, it may be necessary for the system to execute actions that will ensure that the selected schema's preconditions are met. Once they are met, each of the schema's actions are executed in sequence. Thus, each action class in the option specification relates to an action in the schema specification. The outcome of these actions (a unified template) is the production of a short bit of text for inclusion in the story.

Each time a new utterance is produced, a new internal state is realized. In this setting, the internal state represents the production of the utterance—a process we have engineered to complete without failure. Thus, just like the action selection probabilities, each of the internal state transition probabilities $P(\sigma_i^u|\sigma_i^a) = 1.0$.

The implications of this are significant. In effect, there is only one path through the space of actions and internal states defined by the option: $\sigma = u_0 : a_{1,1} : u_{1,1} : \ldots : a_{n,1} : u_{n,1} : e_n$. Therefore, the sum over all paths in Equation 29 is simplified to sum over the only possible path—reducing it to the computation of Equation 28.

What remains is to determine the probability of a story event given that the option has completed its policy (*i.e.*, $P(e_n|\sigma_n^u)$). In terms of influence schemata, this probability is an indicator of how frequently the player is likely to be drawn to make the decision the DM has applied influence to. These probabilities are exactly $P(t : e_n|o, t)$—the probabilities hand-estimated by authors in the original TTD-MDP formulation of the DODM drama manager. Thus, the use of influence schemata as options requires the same author's estimate of player reactions to drama manager actions as the original DODM formalism; however, because we are basing our models on published concepts from social psychology, we have an advantage. In Section 5.5 below, we will derive a method for calculating these probabilities given data from other experiments—a method for transfer of probability models between domains.

5.5 A Method for Cross-domain Transfer of Probabilities

In order for the goal selection algorithm to execute, a probabilistic transition model $P(t : e_n|o, t)$ is necessary. In all of the existing work on TTD-MDPs (and more generally on the DODM formalism) starting with the original formulation due to Weyhrauch [135], the transition model was hand-authored based on a "best guess" of player behavior. This best guess is implemented as a uniform distribution over legal successive story events. When actions are applied to a particular story event, the weight on that event is increased and the distribution is recalculated by normalizing the weights.

More recently, Sullivan *et al.* have examined two other types of player models that encapsulate "world knowledge" about the story environment [121, 122]. Based on the Manhattan distance (or L_1 norm), these models attribute *a priori* weights to plot events based on the physical distance the plot events are from the player in the story environment. In order to extract probabilities, these weights are normalized as with the uniform approach. The effect of an applied drama manager action is somewhat ambiguous in the discussion by Sullivan *et al.* [121, 122]. The basic approach could be one of two methods: 1) arbitrarily increase the weight of the acted-upon story event as in earlier work; or 2) if the action moves the plot event trigger closer to the player, the weight will increase accordingly.

By using social psychology concepts to guide the implementation of our DM actions as influence schemata, we have an advantage over earlier approaches: there is a vast store of literature describing evaluations of the effectiveness of influence methods in various settings; however, one significant question still remains: given data from a real-world (or other story world) experiment, what does it tell us about the probabilistic transitions in another domain? In this section, we will derive formulae to answer that question.

Specifically, we want to be able to derive some transition probabilities from others to estimate the effect of applying influence. The challenge is to be able to do this from the type of data available on the effects of influence.

To make things concrete, suppose $A, B, C, \ldots$ represent potential outcomes and we use a subscript 0 as in A_0 to denote the outcome in the control situation and 1 as in A_1 to denote the outcome under a treatment condition. We will use the terminology *alternative* to mean an outcome with a treatment and we will use letters X and Y when we deal with alternatives and don't specify whether this involves a control or treatment condition. We will use the letters A, B, C to represent outcomes before we specify the control or treatment condition. We would like to be able to go from two outcomes neither of which is more likely to be preferred to the other in the control situation to the probability that we prefer the first outcome to the second when influence is applied to the first. More generally, we would like to be able to calculate the probability that $P(A_1 > C_0)$, interpreted as the probability that outcome A under influence is preferred to outcome C if no influence is applied, if we know $P(A_0 > C_0)$, *i.e.*, the probability that outcome A without influence is preferred to outcome C without influence. This would give us a way to estimate the effect of applying influence to an outcome.

We will discuss how to obtain such probabilities based on the types of data available in the literature. To obtain these probability estimates using the model we present here, we have to make the following assumptions: 1) the model of utility and preference we base our work on accurately describes how people choose between alternatives; 2) a particular technique for influence will for which we know the effectiveness for two alternatives will have a similar magnitude effect when applied to two new, and likely unrelated, alternatives; and 3) authors or technologists will be capable of providing us with "baseline" probabilities indicating preferences between alternatives when no influence is applied.

5.5.1 Types of Data Available

Over the years, there have been countless social psychology studies published that describe the effects of the application of influence to real-life situations. The results of those studies are generally reported in one of two forms: 1) numerical data; and 2) probabilistic data. Often times both types of data are accompanied by summary statistics such as χ^2, mean, variance, etc.

The numerical data experiments report findings based on quantities or scales that are directly measured. For example, Folkes *et al.* present an experiment measuring the effects of product scarcity on usage [39]. Specifically, they conducted three experiments where participants were given a measured amount of shampoo in various sized containers. The results of the experiment indicate that the more scarce the product is perceived to be, the more of it is used by the study participants (*e.g.*, 500 ml of shampoo in a 1,000 ml bottle leads to 87 ml of use on average whereas 250 ml of shampoo in a 1,000 ml bottle leads to 121 ml of use [39]). (This increased use of a scarcer resource is, as the authors note, somewhat surprising.)

Another example of numerical data is that of Regan [100]. Regan examined how reciprocity in the form of a favor can lead to increased levels of compliance. Regan's measurements were made in terms of the quantity of lottery tickets purchased under the different conditions. Therefore, when he reported that in the base condition study participants bought on average 1.00 lottery ticket whereas participants in the influence condition bought on average 1.91 lottery tickets, Regan was able to show a significant effect that reciprocity had on quantity.

On the other hand, data are sometimes reported using frequencies or probabilities. In those settings, the effects of the different conditions in the studies induce a probability distribution over outcomes (sometimes referred to as alternatives), or give us a way to obtain the probability that one outcome is preferred to another. This type of data reporting is a more natural fit with the probabilistic transition model required for DODM and, in fact, our approach will be to translate quantity data into frequency data. To make this more concrete, consider the effects of reciprocity discussed by Cialdini [26]. He reports that the Disabled American Veterans Association gets a response rate of approximately 18% when soliciting donations via a mass mailing campaign. When reciprocity is invoked via an unsolicited gift being included in the mass mailing, the donation response rate rises to 35%. This is an example of probabilistic data reported in literature.

Another example of probabilistic data is reported by Cialdini *et al.* in their study of the effect of reciprocal concessions on compliance with requests for volunteers. In that case, it was found that a mere 16.7% of study participants agreed to volunteer in the control condition, but when reciprocal concessions were employed 50.0% agreed. Similarly, the results of one of our studies discussed in Chapter 7 and Appendix C [110] yield similar frequency data that can be used as input into the domain transfer model presented here.

5.5.2 The Strict Utility Model for Numerical Data

When data is given in terms of quantities rather than frequencies or probabilities, some models allow us to compute frequencies or probabilities. For example, in 1929, Zermelo proposed what has come to be known as the strict utility model. This model describes probabilistic choice in a forced-choice pair comparison system, where for every pair of alternatives X and Y, each trial asks a subject to decide if they prefer X to Y or Y to X, with no indifference allowed. Then $P(X > Y)$ represents the frequency with which (the probability that) X is preferred to Y. We say that a pair comparison system satisfies the ***strict utility model*** if and only if there is a function f that satisfies:

$$P(X > Y) = \frac{f(X)}{f(X) + f(Y)} \tag{30}$$

Here, f is thought of as a utility function over the alternatives.

This model will form the basis upon which we transfer numerical results from one domain to probabilities of player choice in another domain. Let us consider the example from Folkes *et al.* discussed above [39]. In those results, there are five different conditions for which data are presented; however here we will focus on two of them: A_0 and A_1 which according to our notation represent a control (A_0) in which no influence is applied and treatment condition (A_1) in which influence influence is used. Suppose that the data reported for each of those outcomes is given by a function q such that $q(A_0)$ is the quantity reported for outcome A_0, $q(A_1)$ the quantity reported for outcome A_1, etc. In addition to assuming the strict utility model, we make an assumption that the utility of an alternative X is proportional to the quantity reported for that alternative: $f(X) = \lambda \cdot q(X)$.

We are interested in what $q(A_0)$ and $q(A_1)$ tell us about a player's probabilistic choice in a different domain. Specifically, we will show that based on the quantity data $q(A_0), q(A_1)$, if we know $P(A_0 > C_0)$ for some C, then we can derive $P(A_1 > C_0)$. Suppose that

$P(A_0 > C_0) = p$. According to the strict utility model we have:

$$p = P(A_0 > C_0) = \frac{f(A_0)}{f(A_0) + f(C_0)} \tag{31}$$

$$\implies f(C_0) = \frac{f(A_0)}{p} - f(A_0) \tag{32}$$

Thus, we have:

$$P(A_1 > C_0) = \frac{f(A_1)}{f(A_1) + f(C_0)} \tag{33}$$

$$= \frac{f(A_1)}{f(A_1) + \frac{f(A_0)}{p} - f(A_0)} \tag{34}$$

$$= \frac{q(A_1)}{q(A_1) + \frac{q(A_0)}{p} - q(A_0)} \tag{35}$$

This therefore allows us to calculate $P(A_1 > C_0)$ given that we know $q(A_0)$ and $q(A_1)$ from the literature and we either know from the literature or make an assumption about $P(A_0 > C_0)$. Thus, we can construct a probabilistic transition model based on the assumption of control condition probabilities.

For this model and a given method of influence, we have two known values, one unknown value, and a value supplied by authors or technologists. For two alternatives A and B, we know the probabilities that A is preferred to B in the source domain both with and without influence applied that are specified by $P(A_0 > B_0)$ and $P(A_1 > B_0)$ respectively. In practice, these values can be mined from published literature reflective of the specific influence method or from previous experiments. The author or technologist supplies as input to the model the base probability $P(A_0 > C_0) = p$ indicating the preference of A over C they expect to see in the target domain. This value could also be obtained from earlier experiments as well. Using these three values, we can compute $P(A_1 > C_0)$ which is an estimate of the effectiveness of influence in the target domain, subject to the three assumptions described above.

To reconnect this notation with that used to describe DODM, let's assume the episode is in a trajectory t with subsequent events e_1 and e_2. Also, assume we have two options (influence schema) available, one applied to e_1 which we will represent as o_1 and the other applied to e_2 represented as o_2. As always, there's the "do nothing" or "null" option $\emptyset$. Then, we have

$$
\begin{aligned}
P(t : e_1|\emptyset, t) &= P(A_0 > C_0) \\
P(t : e_2|\emptyset, t) &= 1 - P(A_0 > C_0) \\
P(t : e_1|o_1, t) &= P(A_1 > C_0) \\
P(t : e_2|o_1, t) &= 1 - P(A_1 > C_0) \\
P(t : e_1|o_2, t) &= 1 - P(C_1 > A_0) \\
P(t : e_2|o_2, t) &= P(C_1 > A_0)
\end{aligned}
$$

This is then a fully specified transition model. Note that the method described above gives the first four probabilities. To get the fifth and sixth, the method would have to be applied a second time.

To see how this method might be applied, here is an example taken from [100]. We have $q(A_0) = 1.00$ and $q(A_1) = 1.91$. Then, suppose that C is another outcome for which we have no reason to think that either it or A would be preferred in the control situation. Thus, we can assume that $P(A_0 > C_0) = p = 0.5$ in the new domain (as there is no information to the contrary). We have:

$$
\begin{aligned}
P(A_0 > C_0) &= p = 0.5 \\
P(A_1 > C_0) &= \frac{1.91}{1.91 + \frac{1.00}{0.5} - 1.00} = 0.656
\end{aligned}
$$

This gives us an estimate of the effect of influence.

Although this method has been demonstrated on two-outcome alternative scenarios, we have a model for the general case with an arbitrary number of outcomes; however, because the evaluation environment presented in Chapter 7 is a two-alternative environment, the general model is not presented in this dissertation.

5.5.3 The Fechnerian Utility Model for Probabilistic Data

In certain cases, the data available in the literature are presented as frequencies or probabilities rather than quantities and therefore cannot be interpreted as (proportional to) utility estimates. In such cases, the strict utility model (Equation 30) does not apply. Instead, we turn to a more general model of forced-choice pair comparisons known as the ***Fechnerian utility model*** [37, 66]. The Fechnerian utility model holds if there is a monotone increasing function $\phi : \mathbb{R} \rightarrow \mathbb{R}$ so that for all outcomes X, Y,

$$P(X > Y) = \phi[f(X) - f(Y)] \tag{36}$$

Here, as before, $f(X)$ is interpreted as the utility of X. Therefore $P(X > Y)$ is a function of the difference between the utility of X and the utility of Y.

Often times it assumed that ϕ is a cumulative distribution function and $P(X > Y)$ is then interpreted as the probability that X has higher utility than Y. Thurstone's early work on the topic [128, 129] made the assumption that ϕ followed a standard normal (Gaussian distribution with $\mu = 0, \sigma^2 = 1$):

$$P(X > Y) = \phi[f(X) - f(Y)] = \int_{-\infty}^{[f(X)-f(Y)]} N(x)dx$$

More recently, Guilford [43] and Luce [65] proposed that the logistic distribution was a better model:

$$P(X > Y) = \phi[f(X) - f(Y)] = \frac{1}{1 - e^{-[f(X)-f(Y)]}} \tag{37}$$

Note that

$$\phi[x] = \frac{1}{1 - e^{-x}} = p \tag{38}$$

$$\implies x = -ln\left(\frac{1}{p} - 1\right) \tag{39}$$

Assuming the Gilford-Luce special case of the Fechnerian utility model, we derive a method for transferring probabilistic data from one domain to another. Taken from existing literature or another source, suppose that we know the probability that an alternative X is preferred to an alternative Y when influence is applied and when it is not. Specifically, assume we know $P(A_0 > B_0)$ and $P(A_1 > B_0)$. We have

$$P(A_0 > B_0) = \phi[f(A_0) - f(B_0)] = \frac{1}{1 - e^{-[f(A_0) - f(B_0)]}} \tag{40}$$

$$P(A_1 > B_0) = \phi[f(A_1) - f(B_0)] = \frac{1}{1 - e^{-[f(A_1) - f(B_0)]}} \tag{41}$$

Let $\alpha = f(A_0) - f(B_0)$ and $\gamma = f(A_1) - f(B_0)$. Note that we know these two values from Equation 39, Equation 40, and Equation 41. Suppose we have one more piece of information, namely the probability $p \in (0, 1]$ that a player prefers A_0 to some other outcome C_0 without the use of influence. That is $P(A_0 > C_0) = p$. Further, let $\beta = f(A_0) - f(C_0) - \alpha$, so $\alpha + \beta = f(A_0) - f(C_0)$. Note that we know β since we know α and the Guilford-Luce version of the Fechnerian utility model gives us $\alpha + \beta$.

Now we have

$$\begin{aligned} P(A_1 > C_0) &= \phi[f(A_1) - f(C_0)] \\ &= \phi[f(A_1) - f(B_0) + f(B_0) - f(C_0)] \\ &= \phi[\gamma + \beta] \end{aligned}$$

Since we know the value of γ and β, we now know the effectiveness of using influence in the target domain, *i.e.*, $P(A_1 > C_0)$.

Note that the above method works for the general Fechnerian utility model. It is just not as easy to calculate x from $\phi[x]$.

Here is an example using data discussed by Cialdini [26]. We have $P(A_0 > B_0) = 0.18$ which implies $\alpha = f(A_0) - f(B_0) = -1.51635$. Additionally, we have $P(A_1 > B_0) = 0.35$, which implies $\gamma = f(A_1) - f(B_0) = -0.61904$. Let us find an outcome C so that in the control situation, we have no reason to prefer either A or C, *i.e.*, so that $P(A_0 > C_0) = p = 0.5$. Then we have $f(A_0) - f(C_0) = 0$ and, further, $\beta = f(A_0) - f(C_0) - \alpha = 1.51635$. Therefore,

$$P(A_1 > C_0) = \phi[\gamma + \beta] = \frac{1}{1 + e^{-[0.89371]}} = 0.71040$$

which makes intuitive sense.

To convert these probabilities to a transition model for DODM, the same process used in the previous section can be used. The details are left as an exercise to the reader. In Table 25 in Appendix D we present data obtained during one of our user studies that charcterizes the accuracy of this model.

5.5.4 Concluding Thoughts on TTD-SMDPs

In this chapter we have presented a model of TTD-SMDPs to generalize the TTD-MDP formalism from Chapter 3 to handle non-atomic actions. The development of this model was motivated by the influence schemata presented in Chapter 4. In deriving this model, we were able to illustrate how under the right conditions (namely authoring influence schemata for our choose-your-own-adventure storytelling environment) authoring TTD-SMDPs can be reduced to authoring TTD-MDPs. Further, after showing that the authoring problem for TTD-SMDPs can be reduced to the problem of authoring the transition model, we derived a method for transferring a transition model from one domain to another that was inspired by results from the mathematical psychology literature. Lastly, we provided examples illustrating how the domain transfer model is used and produces intuitive results.

CHAPTER VI

SIMULATION RESULTS FOR TTD-MDPS

In this chapter, we present results that highlight the solution quality and performance characteristics of the three solution techniques for TTD-MDPs described in Chapter 3 (Fast Linear Algebra Approximation, L_1 Optimal, and KL Optimal). These results help to characterize the performance of our solution to the goal selection drama management problem (see Section 1.2). All of the experiments discussed in this chapter are simulations. We opted to run these experiments for a few reasons. First, earlier work on the DODM formalism used simulation experiments to characterize algorithm performance and we felt initial comparisons to earlier work were important. Second, in reasonably large domains, it is nearly impossible to obtain enough samples of real traces to get an accurate picture of algorithm performance without the use of simulations. We felt it was important to first tune our algorithms in environments where we could collect hundreds of thousands of samples with only some background processing time. Lastly, we felt simulations would enable to us to construct different test environments that would highlight different performance characteristics of our algorithms. Therefore, the results presented in this chapter were obtained from experiments in three domains: 1) synthetic grid worlds; 2) TTD-MDPs for *drama management* [93, 96, 112, 108, 135]; and 3) TTD-MDPs for adaptive museum tour guides [21, 109].

The results presented in this chapter will highlight the following:

- TTD-MPDs are effective at shifting story quality according to the author's target distribution with a simulated player
- The sampling approaches presented in Section 3.4.1 do have performance differences

- The effects of Gaussian width (variance) are somewhat surprising when using the prototype-distance target distribution paradigm and the number of prototypes used is less important than the relationship of those prototypes
- The solution quality relative to L_1 error and KL error is empirically the same regardless of the method used in practice
- There are differences in the runtime performance of the three methods, but all three are fast enough for real-time interactive systems
- Results on the adaptive tour guide domain provide intriguing evidence that the self-agency-preserving influence models we will discuss in Chapter 7 can result in fewer bad experiences for players while maintaining the majority of good experiences

6.1 The Domains

To better understand the performance tradeoffs of the three algorithms, we selected three domains with varying degrees of complexity ranging from a grid world with a high degree of symmetry and deterministic actions, to a moderately-sized drama management domain originally studied by Nelson & Mateas [93], and to a contrived, but more complex grid world-based museum example domain [21, 109]. In this section, we will describe each of the domains in detail and present the results from a number of experiments. Unless otherwise noted, it should be assumed that the solution method used was the KL-optimal approach described in Section 3.3.3.

6.1.1 Grid World

The "toy" problem that we chose to use to illustrate some of the characteristics of TTD-MDPs and the various solution techniques is a simple grid world [15]. It is a square grid with the start state at the bottom left and the goal state at the top right. There are two actions available to the agent, "move right" and "move up". This is the simple grid world depicted in Figure 3 and discussed briefly in Section 3.2.1.

6.1.2 Drama Management

The specific simulated drama management domains we use in this evaluation are called *Anchorhead* [93, 96] and *Alphabet City* [112]. We selected these two domains in part because they have been used for previous evaluations and because they are of differing size. Anchorhead is a relatively large domain with 29 plot events and 90 drama manager actions. On the other hand, Alphabet City is a bit smaller with only 9 plot events and 28 drama manager actions. In both cases, the transition model used as input for the DODM manager was hand-authored in the style first used by Weyhrauch [135]. It was assumed the simulated player would behave in a uniformly random manner. To model this, the player model was constructed by assigning equal weight to every possible available story event in general. When the DM took an action, the effect would be to modify the weight of the plot point it operated on (either multiplying it by a positive constant or by zero in the case of a deny action) and the new transition model would be computed by normalizing the weights.

6.1.3 Museum Tour Guides

As shown in Figure 8, our museum is modeled by a 4×5 grid of rooms that may contain objects of particular interest with walls preventing some transitions. The transition model permits transitions in every direction provided a wall does not impede. This is in contrast to the simple grid world from above where all transitions are either up or to the right. A trajectory through this grid world is a sequence of rooms modeling a particular tour. **S** is the start room of all trajectories through the museum, while **G** (the gift shop) is the end room. We also model the visitor capacity of rooms in the museum. When above capacity, a room becomes congested. Here, we assume that the agent can detect the current room and can communicate with other nearby guides to determine whether surrounding rooms are congested. Thus, a tour is represented by a sequence of (x, y, c) coordinates that indicate the rooms visited and whether they were congested during the visit. Congestion in a museum corresponds to a player "breaking" a game in an interactive entertainment

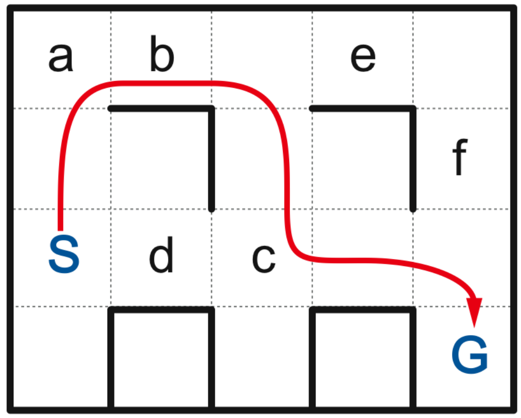


Figure 8: We model a museum as a grid world with walls that prevent some transitions. **S** is the start room of all trajectories (tours) through the museum, while **G** (gift shop) is the end room. The arrow from **S** to **G** shows one of the desired tours. The letters **a**, **b**, **c**, **d**, **e**, and **f** are goals that a museum visitor may have.

setting. That is, when a player's self-agency puts them in a part of the game space that is undesirable to the author.

We represent the congested state of a room's neighbors as a *configuration* of binary variables, $C = \{n, e, s, w\}$ indicating the congestion state (either congested or not) of all four possible neighboring rooms to the north (n), east (e), *etc.* We could consider configurations to be a part of the state space; however, there are two problems with this approach. First, it is unclear why one would want to construct a tour that depended directly upon how crowded neighboring rooms are. Second, such a scheme would require repeatedly solving a system of 2^7 linear equations to obtain the TTD-MDP policy. Alternatively, we choose to treat configurations as observations and condition on them in Equation 2 (restated here as Equation 42 for ease of reading). The resulting equation is Equation 43. This allows us to deal with a system of at most four equations. In this case, we have:

$$P(t') = \sum_{\forall a \in \mathcal{A}_t} (P(t'|a, t) \cdot P(a|t)) \cdot P(t) \tag{42}$$

$$P(t') = \sum_{\forall a \in \mathcal{A}_t} (P(t'|a, t, C_t) \cdot P(a|t, C_t)) \cdot P(t) \tag{43}$$

where C_t corresponds to the congestion configuration of the room that trajectory t terminates in.

In the museum domain, we use the prototype-distance authoring paradigm for the target distribution described in Section 3.4.2. The prototypes represent a set of "good" tours—the museum curator's goals for traffic flow and visitor experience. We use *Levenshtein distance* or *edit distance* as our distance measure.

We are concerned with two kinds of distances between trajectories: "room distance" and "congestion distance." We define *room distance* to be the edit distance between trajectories t and t' defined over the sequence of rooms (represented as their (x, y) coordinates). Similarly, we define *congestion distance* to be the edit distance defined over just the congestion indicators of the trajectories. For example, if two trajectories t and t' visit the same rooms in the same order, but t visits only uncongested rooms while t' visits three congested rooms, then the congestion distance is three.

We assume that different visitors have different goals for their tours of the museum. We account for both naïve and informed visitors. The *naïve visitor* represents a tourist who does not have a particular preference for any of the museum's exhibits aside from what they may have read in a guide book. The *informed visitor* represents a more dedicated art spectator, likely with a larger set of goals.

We explicitly model the destination goals of each of the classes of visitors. These goals represent artworks that are particularly interesting to a class of visitors. The union of the sets of goals of all informed visitors is a strict super set of the union of the goal sets of all naïve visitors. In addition, we consider two variants of these visitor types, for a total of four visitor models. These variants are the *new visitor* and the *returning visitor*. We model new visitors as having no history of satisfied goals, while returning visitors have some percentage of the possible goals already satisfied (we use 35% in our experiments). Note that the history of satisfied goals is only used for evaluating the quality of experience after experimentation—the tour guides are unaware of any goals that may have been met on a previous visit but haven't been met during the current tour. In our 4x5 museum world from Figure 8, we select 10 out of the 20 rooms to contain potential goals for the informed visitor and six to contain potential goals for the naïve visitor. Figure 8 shows the six possible goals used in our experiments with naïve visitors. For each of the visitor types, we randomly assign three goals to be "hidden" goals, or goals that the visitor will enjoy but does not know to pursue.

Each instantiation of a visitor type in an experiment receives some subset of that type's potential goals as their individual goals. We consider two different scenarios where the density of goals in the museum is varied to explore how the layout of goals in the museum affects both congestion and the frequency with which visitors realize their goals. We consider both a *low goal density* and a *high goal density* (where the number of potential goals is doubled).

The tour guides lead visitors by suggesting actions for them to take according to the distribution obtained by solving a TTD-MDP. The available suggestions are $\{north, south, east, west, no\_suggestion\}$. We construct a transition model where visitors usually move toward a known goal location when they are close to it, regardless of the guide's suggestions; otherwise, visitors are more likely to follow the guide's suggestions. Further, visitors prefer not to revisit rooms whenever possible. This model of providing gentle guidance to a visitor is similar to the hint model used in *Declarative Optimization-Based Drama Management* [96].

We divide visitors' willingness to follow suggestions into three categories: those who *possibly*, *probably*, or *definitely* will follow tour guide suggestions. These different categories represent three of the four levels of autonomy we consider (the fourth being a visitor who always *ignores* suggestions). We model these categories of visitors by varying the visitors' transition probabilities.

6.2 *Variety of Experience and Sampling Comparison*

We report on experiments designed to illustrate the overall performance characteristics of the two authorial idioms discussed in Section 3.4 as well as show how some of the variations of each perform. As TTD-MDPs were originally developed for drama management, we evaluate the approaches on the two drama management domains *Anchorhead* and *Alphabet City*. Because existing work has already indicated the potential for sampling approaches to be effective, we choose to simply highlight the relationship between MCMC and uniform sampling (rather than provide a detailed study of sampling performance), and we instead focus the bulk of our attention on experiments in the prototype-distance idiom. In those cases where we evaluate the prototype-based approaches, we ignore hand-authored models to avoid skewing the results too much by our particular choice of prototypes.

6.2.1 Measuring Success

The results in this section are presented in the form of a story quality histogram. The x-axis indicates the quality of a trajectory as defined by the evaluation function and the y-axis indicates the frequency with which trajectories in the sampled set evaluated to that quality. Such histograms have been used for qualitative analysis of drama management systems in earlier work [135, 93, 96]. In these figures, we examine three different techniques: uniform sampling, MCMC sampling, and sampling with SAS+ recovery [112]. The key `nodm` refers to stories for which no drama management was applied and is used as a baseline for assessing the effect of applying the TTD-MDP policy. Of interest in these plots is the relative shape of the histogram curves. The target distribution is constructed so that more highly rated stories are targeted more frequently and so that a variety of stories have non-negligible target probability mass as well. Qualitatively, the goal of the drama manager is to shift the distribution "right and up" (increasing the *quality* of stories) while preserving its "width" (ensuring the *variety* of stories).

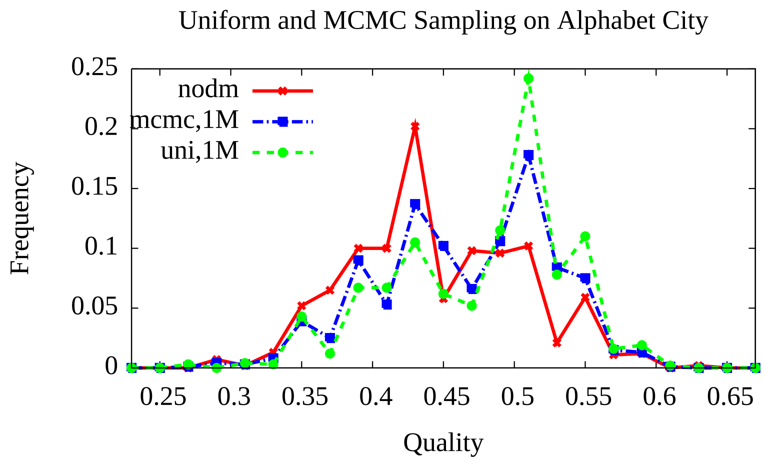


Figure 9: A comparison of uniform and MCMC sampling on Alphabet City.

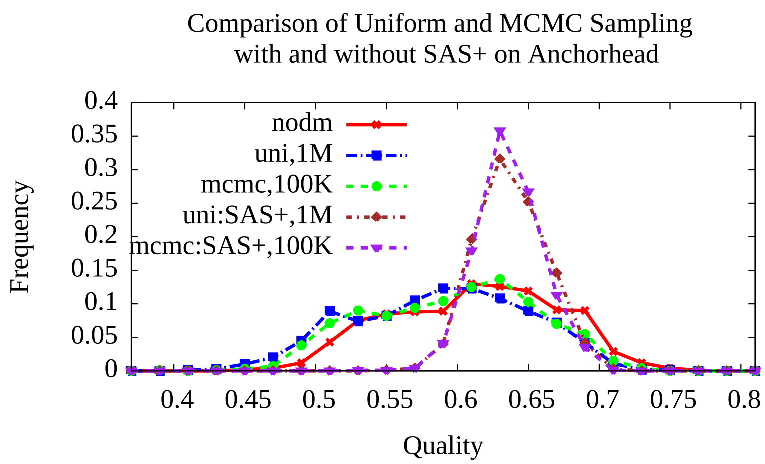


Figure 10: A comparison of uniform and MCMC sampling with and without SAS+ recovery on Anchorhead.

6.2.2 Comparison of Sampling Approaches

First, we discuss the results presented in Figure 9 of an experiment on the smaller *Alphabet City* domain. The three curves in this figure correspond to the `nodm` baseline as well as uniform and MCMC sampling with 1,000,000 samples (and a burn-in of 1,000 in the case of MCMC). The `nodm` baseline is relatively higher toward the bottom end of the evaluation scale and lower toward the top end of the scale than the other two curves. This `nodm` baseline is obtained by simulating gameplay without any DM actions taken, so this result is consistent with our expectations. On the other hand, we found that MCMC tended to perform slightly worse than uniform sampling, as evidenced by the MCMC quality curve being mostly between `nodm` and uniform (*i.e.*, most of the time below `nodm` but above uniform at the bottom of the scale and most of the time above `nodm` but below uniform at the top end). Uniform sampling generally performing better than MCMC will be common to most of the experimental results presented in this chapter. Recall that a distribution is desirable if it tends toward the right and up while preserving width. We believe this relative performance gap occurs as a result of MCMC sampling tending to "hang around" good parts of the story space whereas uniform sampling explores more thoroughly, providing more complete coverage of the trajectory tree. Intuitively, MCMC sampling focuses on parts of the tree in the neighborhood of good trajectories by filling in more of the leaves while uniform sampling balances the samples it generates across the entire tree.

In Figure 10, the results of experiments on Anchorhead similar to those performed on Alphabet City are presented. First, we point out that the `nodm` case slightly beats the performance of uniform and MCMC sampled TTD-MDP policies. This is in contrast to the results obtained on Alphabet City. There is, however, a simple explanation for this difference in performance. Although not presented in detail, the set of actions available to the DM in both story worlds have slightly different characteristics. Most notable is the use of a `temp_denies` action in Anchorhead, where the DM can take an action to temporarily deny a plot point from occurring in the game. At some point later in the game, the DM must reenable that plot point with another action. This would not be a problem for the DM if we could guarantee that the policy were completely specified for every partial story and therefore denied plot events could be reenabled; however, because we construct the policy based on a sampled trajectory tree, there are frequently deviations from that tree before the reenable action can be taken by the DM.

For example, the Alphabet City story world has an average story length of roughly 9 plot events whereas the average story length in Anchorhead is approaching 30. In both cases, the average depth of deviation (*i.e.* number of plot events that occur during an episode before an unsampled part of the trajectory space is encountered) is approximately five. Thus, the Anchorhead domain is at a disadvantage for the following reason: when a plot event is temporarily denied by the first few DM actions, if it is not reenabled before deviation from the tree occurs, then it cannot occur in the story.

To more fully characterize the effect of falling off the tree, in Figure 10 we additionally show the result of using Weyhrauch's SAS+ online sampling search [135]. There are two interesting things to notice. First, the addition of SAS+ significantly improves the story qualities compared to the `nodm` baseline, tending further right and up than the baseline. Additionally, the use of SAS+ results in further right and up distributions when compared to TTD policies without a recovery strategy.[1] Second, the curves are nearly identical, indicating that the deterministic search of SAS+ is able to realize its goals with high probability. This structure in the quality histogram (a steep, impulse-like curve) indicates potential issues for replayability.

6.2.3 Comparison of Prototype-Distance Models

To examine the performance of various prototype-distance models, we conducted a number of experiments to test some of the many free parameters of the approach. In particular, we looked at different distance metrics, different Gaussian widths and different numbers of prototypes. Here we present a selection of the results that are representative of the other experiments we conducted and provide insight into the characteristics of the approach.

Because we are most interested in understanding how the models react to changes in parameters (*e.g.*, changes in how the author specifies the TTD-MDP), we will focus the bulk of our analysis in this section on the drama-management-specific distances (*feature distance* and *blended feature distance*) described in Section 3.4.2. Although not presented

[1] Earlier work has shown that SAS+ alone does not perform well on Anchorhead [96], although a more recent attempt utilizing a pre-cached search tree inspired by our sampled trajectory tree seems to perform more on par with TTD-MDPs with SAS+ recovery [94].

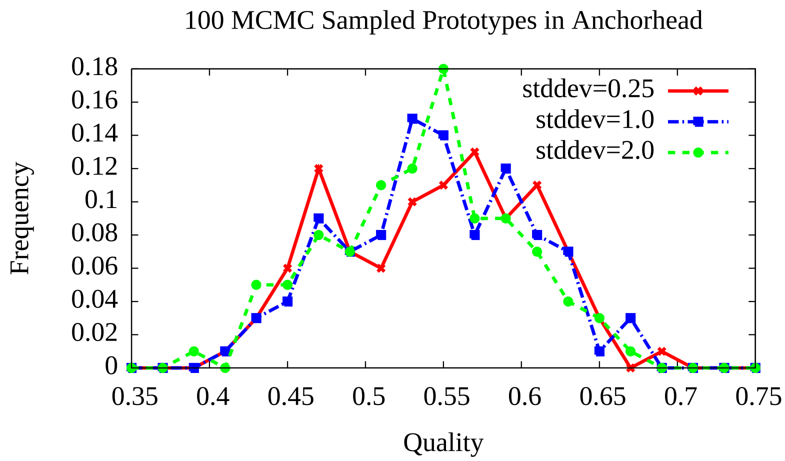


Figure 11: A comparison of models built with 100 prototypes generated by MCMC sampling with various standard deviations on the Anchorhead domain.

here, we have identified similar characteristics in other test domains where the more generic variants of Levenshtein distance and longest common subsequence are more applicable.

In Figure 11 we plot quality histograms for three different prototype models on the Anchorhead domain. The models were constructed with 100 MCMC sampled prototypes after a 1,000 step burn-in and used the feature distance measure. We tested three different standard deviations: 0.25, 1.0, and 2.0. The results we obtained were somewhat counter-intuitive. Specifically, we found that as the width of the Gaussians increased, the width of the resulting quality histogram decreased. We believe the reason for this is related to the idea of a "plateau" in optimization problems. Specifically, with narrow mixture components in the GMM, it is likely that the space between them will have relatively stable and low probability mass; however, as the width increases, one would find that the tails of the Gaussians tend to overlap, forming a nice neighborhood of trajectories that are common to a number of centroids. Therefore, during an episode with small-width Gaussians, if the nondeterminism in the environment causes the current episode to enter the flat space between centroids, the result is likely to end up resembling a random walk through the space. Thus, the quality histogram for experiments with small standard deviation tends to have more mass at the tails. Larger standard deviations do not seem to suffer from this effect.

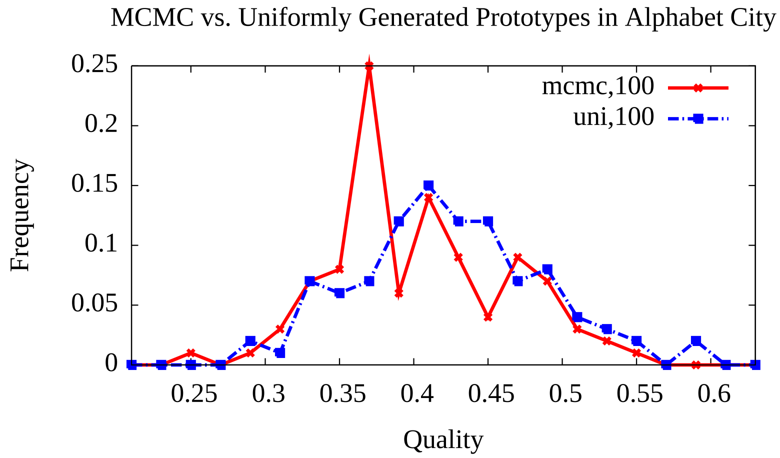


Figure 12: A comparison of models built with 100 prototypes generated by uniform and MCMC sampling on the Anchorhead domain.

Next, we consider the effect that the number of prototypes has on the resulting quality distribution. In Figure 12, we present two prototype models. The prototype models used for this plot were constructed using 100 prototypes after 1,000 sample burn-in and using the blended feature distance measure. As before, a better result will yield a distribution of stories that is further right and up.

We found that the rejection step of the MCMC sampling procedure often leads to clusters of samples, especially when fewer samples are used as prototypes. For example, consider Figure 12. The quality of the MCMC-sampled prototype model is substantially lower than that of the uniform sampled model. Now consider Figure 13, where the same 100-sample uniform model is compared to a 1,000-sample MCMC model as well as a 2,000-sample MCMC model. The quality of the 1,000-sample MCMC model is roughly equivalent to that of the 100-sample uniform model—an order of magnitude increase in the number of prototypes is required for the performance of MCMC sampling prototypes to match that of uniform sampled prototypes. Further, notice how the additional increase in performance obtained by doubling the number of prototypes to 2,000 is noticeable, but not particularly pronounced.

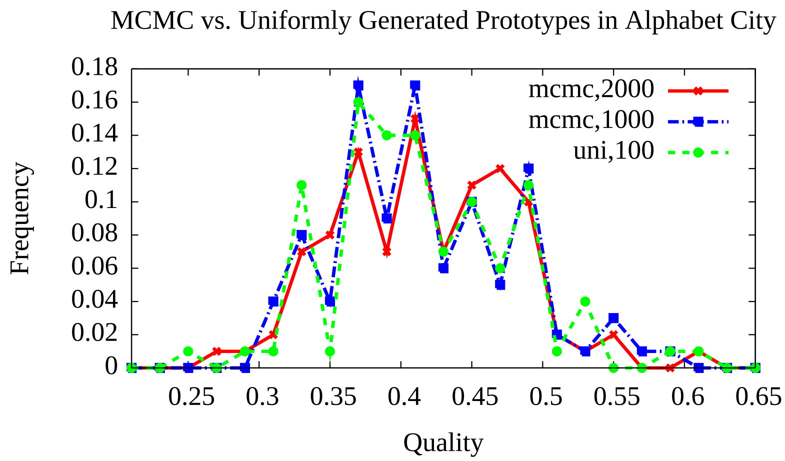
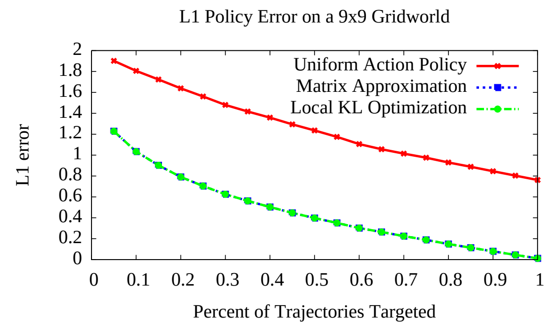


Figure 13: A comparison of prototype models built from 100 uniformly sampled prototypes and 2000 MCMC sampled prototypes on the Anchorhead domain.

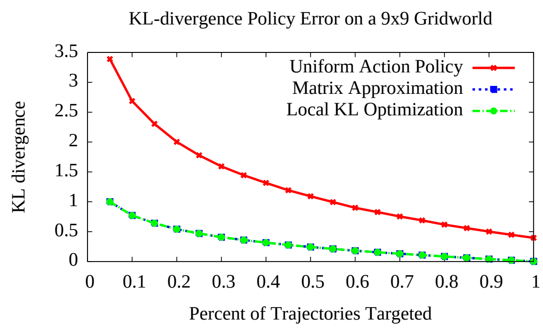
6.3 Solution Quality

To characterize the solution quality of each local computation, we ran a number of trials on the grid world. These experiments are in contrast to the DM experiments which sought to characterize global performance by plotting evaluation quality histograms. Here, the results are reported in terms of *cumulative local error*. To construct the target distribution, every trajectory is enumerated and included in the distribution with probability δ. Each included trajectory was given uniform target probability and the remaining trajectories were given zero target probability. We varied δ from 0.05 to 1.0 in increments of 0.05. We then built the tree and computed the policy according to the local L_1-optimal approach (Section 3.3.2), the local KL-optimal approach (Section 3.3.3), and a baseline uniform policy. We varied the size of the grid from 5×5 to 9×9. For each set of parameters, we ran 10 trials and averaged the results.

The results are presented in Figure 14. Note that we plot error as a function of L_1-error and KL-divergence error because we have derived methods that can provably minimize those measures. As the percentage of trajectories targeted increases, both L_1 and KL error of the local computations decrease. This is expected. Because there is a fixed amount of nondeterminism in action outcomes, as the number of complete trajectories with zero target



(a) Plot of L_1 error



(b) Plot of KL error

Figure 14: Error plots for the 9×9 grid world. The matrix computation and local KL optimization methods perform nearly identically; thus their plots overlap. The maximum difference between the two with respect to L_1 error is 2.48×10^{-6} and with respect to KL error, 9.17×10^{-9}. The uniform action policy is much worse in all cases, by both measures.

probability is higher, the more frequently one of them will be encountered. Thus, reducing the number of zero mass terminal trajectories reduces overall error. Note that the local L_1 and KL approaches perform nearly identically. As expected, the baseline algorithm does far worse than the other two, showing that at least the domain is not trivial.

To test on a computationally intractable problem, we used a version of the Anchorhead drama management domain. With its 29 plot events and 90 drama manager actions, the number of trajectories is astronomically large. We built a sampled trajectory tree to both estimate $m(t)$ by summation and to solve for the TTD-MDP policy. During evaluation it was often the case that nondeterminism in the player model caused an encounter with story trajectories not in the sampled tree. In this event, the fallback policy for the drama manager was to take no action for the remainder of the game. We opt to ignore the recovery

Equation:	$\begin{bmatrix} 0.0027 \\ 0.3586 \\ 0.6388 \end{bmatrix} = \begin{bmatrix} 0.7432 & 0.2535 & 0.5272 \\ 0.1626 & 0.2175 & 0.2172 \\ 0.0942 & 0.5290 & 0.2556 \end{bmatrix} \cdot \vec{\pi}_t$		
Method:	**mat**	**l1-opt**	**kl-opt**
L_1 error:	1.0491	0.5017	0.5017
KL error:	0.7507	0.2875	0.2875
Solution vector:	$\begin{pmatrix} 0.0 \\ 0.0 \\ 1.0 \end{pmatrix}$	$\begin{pmatrix} 0.0 \\ 1.0 \\ 0.0 \end{pmatrix}$	$\begin{pmatrix} 0.0 \\ 1.0 \\ 0.0 \end{pmatrix}$

(a) Example 1

Equation:	$\begin{bmatrix} 0.4177 \\ 0.2182 \\ 0.3640 \end{bmatrix} = \begin{bmatrix} 0.1130 & 0.0025 & 0.0085 \\ 0.6178 & 0.5717 & 0.5559 \\ 0.2692 & 0.4258 & 0.4356 \end{bmatrix} \cdot \vec{\pi}_t$		
Method:	**mat**	**l1-opt**	**kl-opt**
L_1 error:	0.8037	0.7286	0.7991
KL error:	1.1039	0.6444	0.4288
Solution vector:	$\begin{pmatrix} 0.0709 \\ 0.0 \\ 0.9291 \end{pmatrix}$	$\begin{pmatrix} 0.4304 \\ 0.0 \\ 0.5697 \end{pmatrix}$	$\begin{pmatrix} 1.0 \\ 0.0 \\ 0.0 \end{pmatrix}$

(b) Example 2

Figure 15: Two pathological examples where there is a quantifiable difference between the results of the three optimization approaches. The tables contain the equations that represent the local optimization, the solution vector found by each of the three methods, and the error associated with each of the solution vectors.

mechanism considered by Roberts *et al.* [112] and used in the experiments described in Section 6.2.2. We omit this recovery technique in favor of a "pure" comparison of the optimization methods.

Again, the results for the matrix computation and KL optimization techniques were nearly indistinguishable, consistent with what we observed in the grid world. It is possible, however, to construct cases where the matrix computation approach fails to produce the optimal solution. The following examples use 3 actions and 3 successor states. We will use **mat** to refer to the suboptimal linear algebra approximation; **l1-opt** to refer to the correct L_1-based solution outlined in Equation 12 of Section 3.3.2; and **kl-opt** to refer to the KL-based algorithm. We present the results of the three approaches on two pathological examples in Figure 15.

In the first example (Figure 15(a)), **kl-opt** does better than **mat** with respect to L_1 error. In fact, the difference in L_1 error between the two is 0.5, which is 25% of the maximum possible L_1 error (2.0). In the second example (Figure 15(b)), though **kl-opt** cannot beat **l1-opt** with respect to L_1 error (by definition), it *does* do better than both **l1-opt** and **mat** w.r.t. KL error and does equally as well on L_1 error.

In both pathological examples, the local transition matrix makes it very difficult to reach the target distribution. The first row of the transition matrix contains relatively large numbers compared to the target probability mass for the first row of the target probability vector. In all three domains we have examined, this situation does not arise often (if at all). It could be that any local technique that attempts to match the local target distribution according to a reasonable error measure will perform quite well in practice on the sorts of problems we have explored; however, many such techniques will fail in exactly the difficult cases. By contrast, we have derived an algorithm that provides theoretical guarantees and, as we shall see, remains computationally feasible even within the performance constraints of an interactive system.

6.4 Execution Time

Given that both the optimization technique and the matrix computation approach perform similarly in the types of domains we have studied, we seek to characterize the computational tradeoffs that are made when one algorithm is used over the other. The following experiments were run on an Intel Pentium Xeon Processor at 3.8GHz with a 2,048 KB cache and 8GB of RAM (although only 1GB was allocated to the Java 1.5.0 `server` JVM).

First, we examined the runtime of **mat**, **kl-opt** and a uniform random policy on the grid world domain. Table 12 contains the results of those experiments. Note the rapid growth in running time for all of the approaches. This is due to the exponential increase in trajectory tree size as the size of the grid world increases. Specifically, for an $n \times n$ grid world, there are $\frac{(2n-2)!}{(n-1)!(n-1)!} = O(n!)$ local optimizations to be solved. In the case of **mat**, each of the local computations involves solving a system of linear equations, which can be accomplished relatively efficiently—$O(a^3)$ for a naïve implementation where a is the number of actions. On the other hand, the **kl-opt** technique requires a polynomial number of steps.[2]

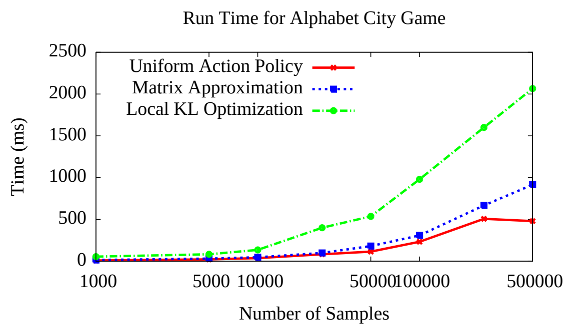
Table 12: Average computation time for the uniform baseline, **mat**, and **kl-opt** algorithms for various grid sizes.

	Total time (ms)			Normalized time (ms)		
Size	**uni**	**mat**	**kl-opt**	**uni**	**mat**	**kl-opt**
5×5	1.90	2.58	76.88	0.0271	0.0369	1.0983
6×6	7.22	8.10	233.16	0.0287	0.0321	0.9252
7×7	34.30	34.52	840.48	0.0371	0.0374	0.9096
8×8	161.02	167.50	3191.48	0.0469	0.0488	0.9299
9×9	769.06	789.70	12119.28	**0.0598**	**0.0614**	0.9417

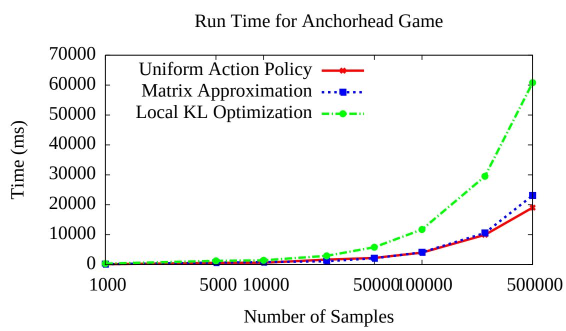
As shown in Table 12, the run time for the **kl-opt** approach is significantly higher than both the linear algebra approximation and the random baseline. To examine this effect more closely, we consider the normalized computation time (*i.e.*, time per local optimization). Note how for the **kl-opt** approach, the normalized running time remains essentially constant for all grid sizes. However, for the baseline and the **mat** approach, the normalized run time jumps significantly for the largest grid size (marked in bold in the table). We attribute this to cache misses. The increased time required per computation for the **kl-opt** approach hides the latency encountered for a cache miss.

In an earlier result of Roberts *et al.* [112], it was shown that the number of samples used to construct the trajectory tree in the drama management domain has little impact on the accuracy of the resulting policy. Thus, we examine the run time performance here to determine if the theoretical guarantee provided by the **kl-opt** algorithm outweighs any increase in complexity. We examined run time performance of the algorithms in the two drama

[2]The polynomial is a function of a number of parameters of the optimization and the condition number of the objective function's Hessian. A complete analysis is beyond the scope of this dissertation.

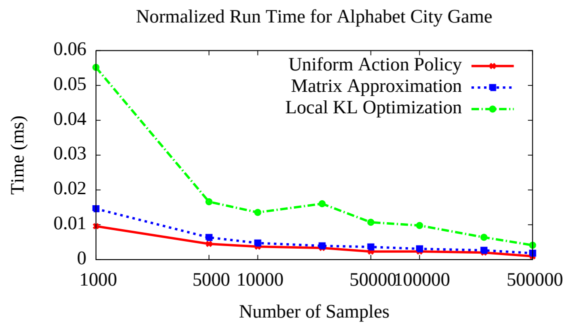


(a) Alphabet City

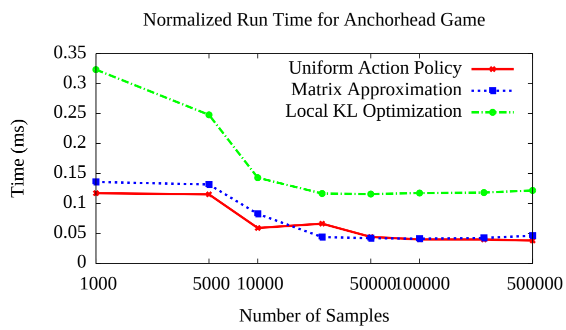


(b) Anchorhead

Figure 16: Average computation time for the *Alphabet City* and *Anchorhead* story worlds based on the number of samples used to build the trajectory tree.



(a) Alphabet City



(b) Anchorhead

Figure 17: Average normalized computation time for the *Alphabet City* and *Anchorhead* story worlds based on the number of samples used to build the trajectory tree.

management domains, Alphabet City and Anchorhead. In Figure 16, we present the average run time for each of the algorithms based on the number of samples used to build the trajectory tree (note the log scale on the x-axis). Unsurprisingly, the average computation time grows linearly with the number of samples for all three methods. However, it would appear that the growth rate for the Alphabet City story (Figure 16(a)) is actually sub-linear.

To verify this behavior, we again present normalized running times for each of the algorithms in Figure 17 (note the log scale on the x-axis). Here, we see that once approximately 25,000 samples are used, the normalized run time remains about constant for the Anchorhead game whereas it continues to decrease for the Alphabet City game. This is likely due to the lower branching factor associated with Alphabet City.

Although **kl-opt** requires roughly thirty times the computation time compared to **mat** per decision point, we are well under what is necessitated by a real-time domain; other researchers have allocated as much as two *seconds* between decision points for a drama manager to select an action [93]. These results indicate that the local **kl-opt** approach can execute in less than a millisecond, easily meeting the constraints of real-time interactive drama management and other systems. Further, this approach retains its optimal properties even when used online, so the increased time per local computation can be made to be only a linear penalty in practice and, therefore, a small price to pay for a guarantee of optimality.

6.5 Results for Museum Tours

We performed a number of experiments to examine how varying levels of visitor autonomy and tour guide control change the quality of the experience in the museum example. We explored two measures of quality and compared results for our TTD-MDP tour guides to three other approaches. The first two of these approaches use no tour guide: 1) *wander*, a visitor who acts randomly, and 2) *ignore*, a visitor who pursues a goal when one step away but wanders otherwise. The third approach, *random*, uses a tour guide that chooses actions uniformly at random.

6.5.1 Setup

We selected a set of prototype trajectories for both the naïve and informed visitors. In these experiments, naïve visitors had two prototype trajectories while informed visitors had three. We assigned the same set of prototype trajectories for both the new and returning visitors of each type. A prototype trajectory must begin in the entrance room and end in the gift shop, and every possible goal must lie on at least one prototype trajectory. The second condition prevents the tour guide from being unduly penalized for not guiding a visitor to a goal when that goal is not available on some prototype trajectory.

We chose uniformly from the four visitor types and allowed them to enter the museum at a constant rate of n per simulation step. We selected a room capacity to reflect the average number of visitors that we expected to be in the museum at any given time step. If this was set too high, then none of the rooms would ever be congested. On the other hand, if it were too low, then all of the rooms would be congested and none of the realized tours would reflect the prototype tours closely. We will present results that verify this empirically.

To model the reality that visitors do not move in lock step, we selected a random ordering over all visitors currently in the museum and allowed them to move to a neighboring room in this order. We updated the congestion state of all rooms after each visitor moves. Once every visitor in the museum had a chance to move, we repeated the process.

We made one additional simplifying assumption about trajectory lengths. We assumed that every visitor makes a fixed maximum number of steps through the museum and then moves directly to the exit if they have not already reached it. In these experiments, we limited tour lengths to 10 rooms. When trajectories became significantly longer than the prototypes used to author the target distribution, we observed endless wandering behavior. This occurred because as trajectory length increases, every subsequent trajectory will appear equally far away from the centroid, regardless of whether or not the visitor is moving toward the gift shop.

6.5.2 Measuring Success

To judge the performance of the TTD-MDP-based tour guides, we wanted to characterize the distribution of realized trajectories of individual visitors. To accomplish this, we looked at the distribution of distances of each tour from its closest centroid. To be more precise, we created histogram bins for each distance $0, \ldots, l$ (where l is the maximum permissible trajectory length). Then, for each tour encountered, we selected the closest centroid and incremented the histogram bin associated with that distance.

To measure how effective our tour guides were at realizing visitors' goals, we considered three statistics: the percentage of a visitor's known goals that were achieved, the percentage of a visitor's hidden goals that were achieved, and the frequency of congested rooms experienced by each visitor.

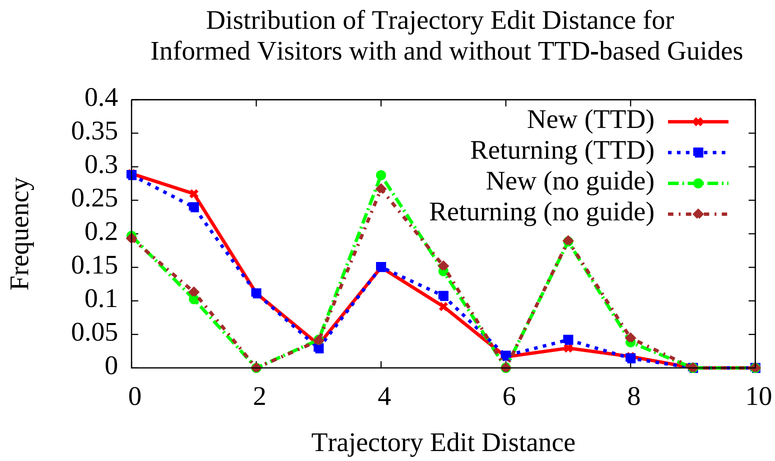


Figure 18: Distribution of Trajectory Room Distance for Informed Visitors with and without TTD-based Guides.

6.5.3 Results

The data for the following experiments was obtained with a low goal density, a room capacity of four visitors, a rate of five visitors added to the museum per simulation time step, and visitors with a fairly low probability of accepting tour guide suggestions (the *possibly* visitor category).

In Figure 18, we plot a histogram of room distance for new and returning informed visitors with and without the benefit of a TTD-based tour guide. Here, the target distribution for the TTD-MDP-based guides was a Gaussian distribution defined over distance from centroids (*i.e.*, a zero-mean Gaussian). Notice the relative shape of the distribution of distances for the trajectories obtained using the TTD-based tour guides—one half of a Gaussian. This illustrates nicely that despite the relative lack of cooperativeness of this visitor type, we still see a distribution over distance that roughly matches the shape of the mixture of Gaussians model—the designer's goals for a distribution over distances from centroids is realized. The data for the informed visitor without the tour guide does not exhibit this behavior. The dips at distance three and six in this plot are attributable to the structure of the museum grid world, the location of potential goals, and the set of prototypes. Specifically, once the visitor enters a particular part of the space (*e.g.*, the top

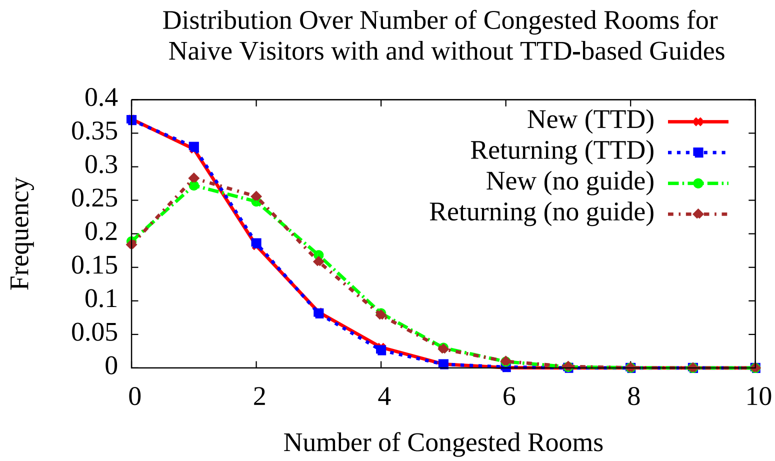


Figure 19: Distribution of Congestion Distance for Naive Visitors with and without TTD-based Guides.

left or top right corner in Figure 8) there are a limited number of locations from which they can diverge to another path, thus making deviation less likely in these regions.

In Figure 19 we plot a congestion distance histogram for new and returning naïve visitors both with and without the TTD-based guides. For this experiment the target distribution was defined as a zero-mean Gaussian over the number of congested rooms. There are two interesting points here. First, the rate of congestion is almost identical for the new and returning visitors in each case. Second, note the relative position of the curves for the trials with and without the guides. Visitors with guides experienced fewer congested rooms, with a histogram peak at 0 congested rooms, instead of at 2 congested rooms for visitors without a guide. Further, the number of visitors who experienced a higher number of congested rooms was far fewer.

6.5.4 Goals and Congestion

In this section, we examine both congestion and goal realization together. We consider these in aggregate for all visitor types. In these experiments, we would like for the rate of congestion to be as low as possible and for the rate of goal realization (new or hidden) to be

Table 13: Aggregate statistics for visitor models with low and high goal density.

Measure	Congestion		New Goals		Hidden Goals	
Model	L	H	L	H	L	H
TTD:	**0.135**	**0.153**	**0.476**	**0.598**	**0.289**	**0.351**
ignore:	**0.209**	**0.202**	**0.497**	**0.608**	**0.290**	**0.374**
wander:	0.517	0.517	0.113	0.271	0.118	0.273
random:	0.287	0.247	0.398	0.554	0.226	0.342

Table 14: Aggregate statistics for visitors that *probably* follow guide instructions with varying room capacity limits.

Measure	Congestion		New Goals		Hidden Goals	
Capacity	L	H	L	H	L	H
inf	0	0	0.501	0.569	0.313	0.346
6	0.021	0.029	0.487	0.564	0.307	0.344
5	0.049	0.062	0.472	0.560	0.297	0.342
4	0.116	0.133	0.441	0.543	0.272	0.338
3	0.244	0.259	0.416	0.534	0.253	0.333

as high as possible. In Table 13 we consider the results of experiments both with and without TTD-MDP-based guides as well as with both the *wander* and *random* baselines. “L” and “H” represent low and high goal density experiments. These results are shown averaged across all visitor models (naïve and informed in both the new and returning variants). The *wander* and *random* baselines do not perform well in any of the categories. In the case of the *wander* baseline, this is attributable to a lack of goal directed behavior. For the *random* tour guide, however, this is attributable to the willingness of the visitor to follow the guide’s random suggestions. In comparison to those baselines, the *ignore* and TTD-MDP-based guides cases yield very promising results. Specifically, we see a noticeable reduction in congestion that accompanies a significant increase in goal realization. Note the differences between the TTD-MDP-based guides case and the ignore case (in bold). There we see that the rate of congestion is greatly reduced while goal realization is preserved when TTD-MDP-based guides are used.

Table 15: Aggregate statistics for visitors with varying willingness to follow suggestions.

Measure	Congestion		New Goals		Hidden Goals	
Model	L	H	L	H	L	H
ignore	**0.209**	**0.202**	0.497	0.608	0.290	0.374
possibly	**0.135**	**0.153**	0.476	0.598	0.289	0.351
probably	**0.116**	**0.133**	0.441	0.544	**0.274**	**0.338**
definitely	**0.091**	**0.090**	0.364	0.450	**0.315**	**0.385**

6.5.5 Capacity and Visitor Autonomy

In Table 14, we summarize the effects of room capacity. Again, we would like for the rate of congestion to be low and the rate of goal realization to be high. In particular, we see that the effects of room capacity on goal realization are more pronounced in the low goal density case than in the high goal density case. As capacity decreases, the percentage of realized goals in the high density case remains essentially the same. The effect on goal realization of decreasing room capacity in the low goal density case is exaggerated because the same number of visitors are sharing a desire to achieve fewer goals. As a result of the guide's tendency to suggest alternates to congested rooms and the visitor's tendency to follow those suggestions, we also see a reduction in goal satisfaction.

In Table 15 we organize the data to consider the effect of autonomy on congestion and goal realization. The data in the table was obtained by varying the visitors' willingness to follow advice. Note how the rate of congestion is slightly lower for the low goal density case. We attribute this to visitors not having as many goals to seek, and therefore being more willing to follow tour guides' advice that may lead them both away from congestion as well as away from unrealized goals. Taken together with the percentage of satisfied goals, this data is informative. We see that the more willing a visitor is to follow the tour guide, the less congestion it will encounter, but the fewer goals it will realize; however, this tradeoff may be worthwhile—a 26.0% reduction in goal satisfaction accompanies a 55.4% reduction in congestion (in the high goal density experiment).

There is an exception. Although the frequency of realization of hidden goals generally decreases as visitors follow their guides, if they always follow their guides they start to realize more hidden goals again. This occurs because the guides have some sense of where hidden goals may be, due to the museum curator's well constructed prototype tours.

As visitors have more autonomy, they achieve more of their goals because of their willingness to ignore the tour guide and pursue a known goal; however, this gives rise to a tragedy of the commons: when visitors always act only in their own immediate interest, they end up in crowded parts of the museum lessening the quality of the experience for everyone. On the other hand, if visitors always listen to the tour guide, they experience less congestion at the expense of realizing fewer of their known goals. Somewhere in the middle of these extremes is a "sweet spot" where visitors exercise enough self-agency to express their desires but listen to the tour guide enough to benefit from the designer's goals.

6.6 Summary of Analysis

In this chapter, we have presented the results of a number of simulation experiments performed to help shed light on the performance characteristics of the various algorithms we developed for TTD-MDPs. While an interesting exercise by themselves, these results were in service of a larger goal: to enable a more thorough understanding of how TTD-MDPs are likely to perform in the face of actual human participants. Before investing the time to evaluate TTD-MDPs for managing an interactive experience with human participants, we wanted to understand how the various alternatives and parameter settings might change the overall behavior. The lessons we learned from these experiments informed the design choices we made for our user studies (presented in Chapter 7) and are the following.

- In the simulated story domains we found:
    - When using a sampling technique to construct an estimated target distribution based on a sampled trajectory tree, uniform sampling appears to provide better ultimate results than rejection sampling (MCMC) due to better coverage of the story space with the same number of samples

- In sufficiently large domains, a sampled trajectory tree alone does not provide enough coverage for good performance and an online recovery technique (such as SAS+ [135]) is required
- Using sampling to create a set of centroids for a prototype-distance target distribution also seems to perform better with fewer samples when uniform sampling is used in comparison to rejection sampling

• In the simulated grid world domain we found:

- While there are no theoretical guarantees that the linear-algebra-based approximation for solving a TTD-MDP will result in an optimal policy, in practice the results appear comparable to that of the provably optimal KL divergence minimization
- Although there is a penalty in terms of computation time to perform the KL optimization, the time required is still well within the constraints imposed by a real-time interactive system

• In the simulated museum domain we found:

- TTD-MDPs are effective with a multi-variate Gaussian mixture target distribution (as compared to the one-dimensional Gaussian mixture used in the story domains)
- Perhaps most significantly, we found that in cases where museum visitors are likely to, but not always willing to, follow the guidance of the TTD-MDP manager, they get a notable reduction in "bad things" while only missing out on a few "good things" (a 26.0% reduction in goals and a 55.4% reduction in congestion). This is encouraging evidence that our approach of using influence to guide (rather than heavy-handed modifications to force) the players' decisions

can benefit the players by steering them away from bad things in a story while giving them the agency to realize the good things they want from the story

CHAPTER VII

USER STUDY RESULTS

In this chapter we will present our choose-your-own-adventure-style storytelling system and describe the results of what is, to our knowledge, the first successful use of social psychology concepts to shape player experiences in an interactive story and first successful implementation and evaluation of a DODM drama manager with system-generated and system-refined actions. The contents of this chapter are devoted to our approach as a whole, combining the solutions to all three drama management problems: goal selection, action/-plan selection/generation, and action/plan refinement (see Section 1.2). The storytelling system we describe here is the platform upon which we evaluated the influence models presented in Chapter 4. We have designed the system to provide some of the characteristics of a full-blown interactive storytelling system, but retain the simplicity necessary to conduct more tightly controlled studies. It does have some complexity which induces some confounding factors in our analysis, but we view these as necessary in order to evaluate our models in an interactive environment.

We will describe the system architecture in detail, as well as present the design of our two user studies: a study with hand-authored influence statements and our end-to-end evaluation of DODM, TTD-SMDPs, influence schemata, and natural language templates. We will present only some of the results of our first study which serve as a comparison to the results of our second study. For readability, we will omit some of the details of the results from the second study as well. The interested reader can find the complete results of the first study in Appendix C and the complete second study results in Appendix D. The first study was conducted using hand-authored influence statements to illustrate that influence can be effective in an interactive environment as well as validate our study design. The results of the study indicated that influence can indeed be effective. Our end-to-end evaluation in the second study confirmed this conclusion.

Although we found influence to be effective, we did have an unexpected result come out of the analysis of the first study data: players presented with influence reported differing levels of feeling control over the experience (see Section 7.3.5 for more details). We hypothesized that story content may have been the cause for this finding in the first study, rather than the use of influence. Results of our end-to-end evaluation have helped to further shed light on this and will be discussed in Section 7.7.2.

Briefly, the findings we will present in this chapter are:

- Properly implemented, influence can be effective at shaping player decisions in an interactive choose-your-own-adventure story
- TTD-SMDPs, combined with influence schemata and natural language templates, are effective at reducing the KL-divergence between a target distribution over stories and a distribution over stories realized by amassing player experiences
- We failed to find any significant effect resulting from the use of influence on a player's sense of self-agency as measured by a Likert prompt and a modified "locus of control" scale
- The use of influence can lead to increased feelings of engagement with the system and connection with the main character as measured by a Likert prompt

7.1 Study 1 Methodology

Here we describe our methodology for the first study. We will discuss the experimental design, how participants were recruited, the story we authored, as well as the type of data we collected.

Our framework for evaluating computational models of influence is a web-based choose-your-own-adventure-style interactive storytelling system. Our system displays a sequence

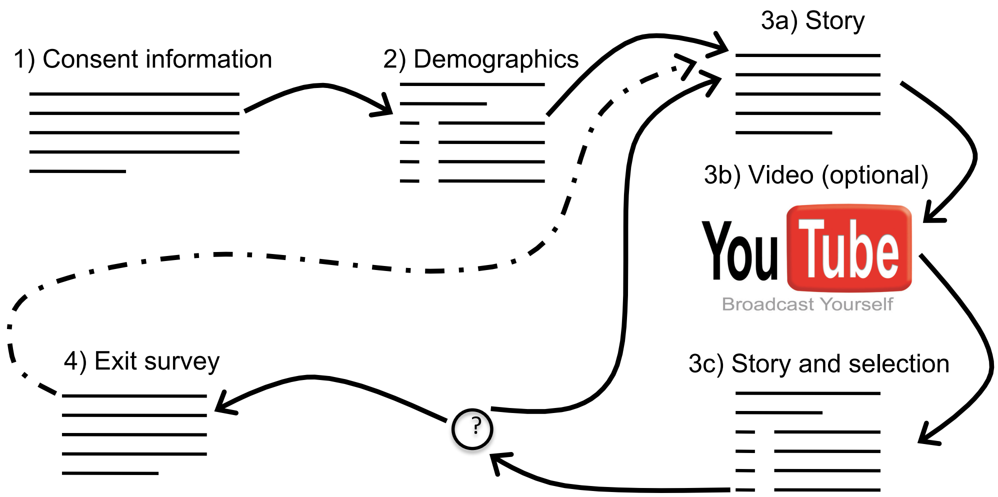


Figure 20: A schematic of the storytelling system. The player begins with the consent information and a brief demographic survey, then iterates through story events consisting of text-video-text-question tuples, and finishes with a brief exit survey. The player has the option to play again.

of authored text (and optionally selected videos) that comprise narrative units, or events, that are linked together by explicit decision points for the player. The videos, when used, were obtained from YouTube[1], a free online repository for streaming video. A player's experience advances according to the procedure depicted in Figure 20 which is as follows:

1) Consent information: Upon arriving at the landing page of our web-based system, the player is presented with the study's consent information and an acknowledgement is obtained before proceeding

2) Demographics: The player is asked to answer a set of basic demographic questions. The data collected from this survey enables a comparison of participant groups. Upon completion, the player is presented with brief instructions for continuing

3a) Story: The player sees author-provided text that describes the beginning of a story event (later referred to as "pre-text")

[1]`http://www.youtube.com/`

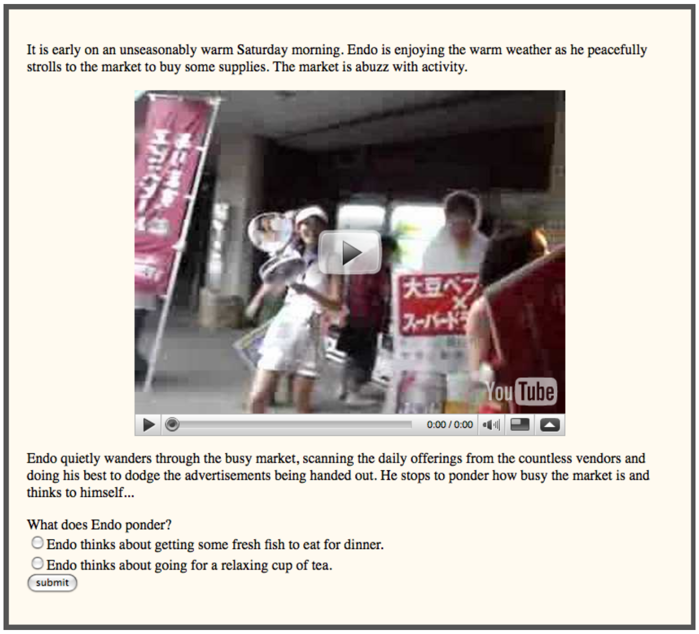


Figure 21: A screenshot of the storytelling system during a story event from the first study. The player sees text containing story information, a video, more text with story information, and a question-response set that allows the player to make choices for the main character.

3b) Video (optional): The player is presented with a video from YouTube providing information supplemental to the story. The video does not start playing automatically; if it is included, it is completely optional for the player to view it

3c) Story and selection: After the video, the player is presented with a short bit of text (later referred to as “post-text”) and a two-alternative multiple choice question. The question solicits a decision from the player that will determine the next story event and therefore advance the narrative. For the purposes of this study, there are always two answers to the questions, but there is nothing in the architecture design that prevents more options. At this point, the system cycles back to Step (3a) and displays another text-video-text-question set or, once the story has reached a conclusion, moves on to Step 4

4) Exit survey: The player is presented with a set of five statements and is asked to indicate their level of agreement with the statements on a Likert scale [63].

All of the text, question-response sets, and videos (when used) are pre-authored or obtained ahead of time—not generated dynamically. We opted to obtain the videos from YouTube to ease the authoring process. In retrospect, the time required to find a video pertinent to the story content was deemed to not be worthwhile. Further, the use of videos does not enable a measurement of anything we are interested in measuring. All of the story events and their components (text fields, questions, answers, and videos) are given a unique identifier that is stored in a MySQL database along with the html code to embed the video and any meta-data we associate with it. The story was presented to the player using lightweight web-programming including css stylesheets and Java Server Pages. Figure 21 is a screenshot of our production system in Step (3) during the first story event of the study.

7.1.1 The Story

The story presented in our first study takes place in Japan. The main character, a young man named Endo, is enjoying a lazy Saturday. What he experiences during the day is, in part, controlled by the decisions the players make for him. The story consists of 24 events (not all of which will occur in any given episode) that are based on 16 YouTube videos, 24 authored bits of pre-text, 18 authored bits of post-text, 14 questions, and 28 answers. There are 44 possible transitions that link the events in various orders. The story always begins with Endo in a market and the subsequent events take place in other parts of the city. Some of the events include going for a cup of tea, shopping for a knife, shopping for a camera, walking in the imperial palace garden, and buying fresh fish at the fish market. The first 3-4 events of the story are all very diverse. The conclusion of the story is always the same regardless of the choices made near the beginning; however, the path to the conclusion can be very different.

7.1.2 Participant Groups: Control vs. Treatment

One of the goals of our studies is to provide data for assessing the claim that influence statements can bring about desired behavior in players. In addition, we were interested in

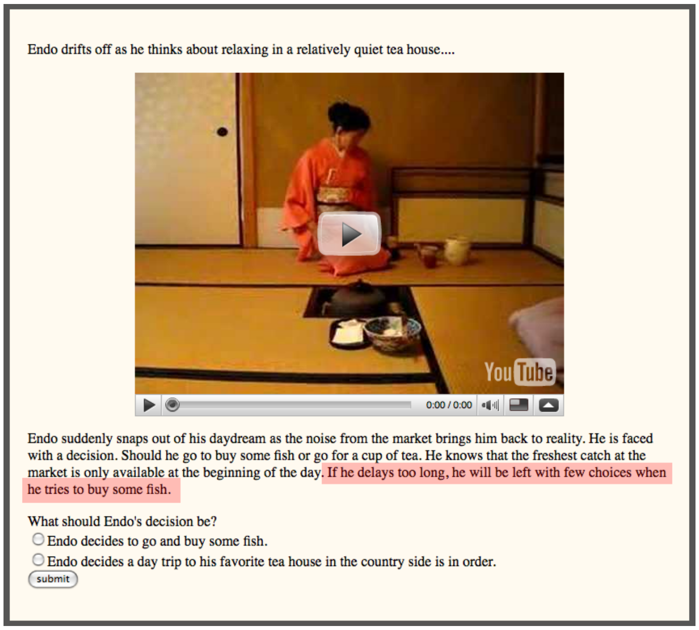


Figure 22: A screenshot of the storytelling system during a story event in which an influence statement (highlighted here) has been included. The presentation of the text, video, and question-answer sets remains the same as for the control group; however, an influence statement in the form of at most three sentences is added to try to shape the player's decision making.

understanding the effect of using influence models on a player's sense of self-agency. In order to characterize these effects, we needed to compare results between populations.

Our study design includes two groups that players are randomly assigned to at the beginning of each episode: a control group that was given both surveys and the base story; and a treatment group whose participants' experience was identical to that of the control group except for the addition of influence statements.

To limit the number of variables changed for the treatment group in the study, before running the study we selected one of the story events as our "goal." In this case, the goal was that the player chooses for the main character to buy fish at the fish market (one of the story events). There were five story events that induced three possible paths the player could take that would result in the goal being realized and four possible paths that would not have. With the exception of the first story event, the remaining four events were associated with a hand-authored influence statement of no more than three sentences in one of four places: before the pre-text; after the pre-text and before the video; after the video and before the post-text; or after the post-text. Figure 22 is a screenshot of the system in the influence treatment where an influence statement has been inserted after the post-text. Note that the influence statement is emphasized in that figure for reference purposes, but appeared as normal text to the player.

To further limit the number of variables changed for the treatment group in the study, we chose to base our hand-authored influence statements on only one of the six principles of influence presented in Chapter 4: ***scarcity***. Here are two examples of the supply and deadline variants of the scarcity principle used: 1) "[Endo] knows that the freshest catch at the market is only available at the beginning of the day. If he delays too long, he will be left with few choices when he tries to buy some fish;" and 2) "While he waits, Endo checks his watch. With each passing minute, the chances of Endo finding a good fish to buy at the market are decreasing rapidly."

7.1.3 The Data: Traces and Surveys

There are two techniques we employ to collect data during our studies: *traces* and *surveys*. Through observation, traces allow us to passively keep track of information about players' experiences while interacting with our system. On the other hand, surveys allow us to actively elicit specific information about the play experience. Both types of data play an important role in our characterization of the results. In this section, we will discuss our collection techniques in detail.

7.1.3.1 Traces

As the path through our storytelling system is sequential, the traces we collect are sequential as well. Every time a player is presented with a story event, it is logged. Further, every time the player confirms their answer choice and moves to the next event, the log entry associated with that player and the particular story event is updated with their answer choice. Further, all entries are annotated with a unique session identifier. This session identifier is assigned to a player when they begin the demographic survey portion of the study and remains active until they browse away. This enables us to keep track of single-session replay counts. We opted to stop short of using browser cookies to maintain session identifiers across multiple visits, citing anonymity concerns for IRB purposes as our motivation. Thus, if a player was to play through the story, then come back the following day and play again, we would not know it was the same player. In addition to session information, we kept track of whether or not influence was applied.

These traces enable us to obtain a lot of information about our environment. For example, we can determine how many players "dropped out" of the study (*i.e.*, didn't finish the story) and how many played multiple times in a row. We can determine the relative frequency of answer choices which serves as a baseline for characterizing the effectiveness of influence.

7.1.3.2 The Demographic Survey

To facilitate a comparison of players' experiences with and without the use of influence, participants in the study were randomly assigned into one of two conditions: a control and an influence treatment. To control for the possibility that any differences in trace information or exit survey responses is a function of population differences between the two groups, we collected demographic information about them. In analyzing responses to the demographic questions we hoped to fail to find a significant difference between the groups of study participants.

We had participants answer four questions pertaining to their education level, age, and the number of hours per week they spend on the internet and playing computer games. Participants were not required to answer the questions; their lack of response was recorded to enable a sensitivity analysis. Education level was broken down into a seven point range from "high school" to "Ph.D." Age was measured by an eight point range starting below

18 years old and ending above 50 years old. And lastly, both internet and game hours were measured on a six point scale beginning with fewer than four hours and finishing with more than 25. We selected these four questions as well as response sets in part based on the types of people we expected to be able to recruit. The complete set of demographic survey questions is included as a reference in Appendix B. The complete distributions of responses are also included for reference in Appendix C and Appendix D.

7.1.3.3 The Exit Likert Prompts

A Likert scale [63] is a psychometric scale commonly used in questionnaires for survey research. Likert scales differ from the more general rating scales in that they ask participants to indicate their *level of agreement* with a prompt, or statement. Likert scales are typically defined by the number of "levels" of granularity they offer. For example, a five-level Likert scale (which we have used in our surveys) asks participants to indicate their agreement with the prompt by choosing from one of the five categories: "strongly disagree," "disagree," "neither agree nor disagree," "agree", or "strongly agree." Typically Likert scales are implemented with an odd number of levels to avoid forcing survey respondents into having to choose agreement over disagreement, although that need not be the case.

Our exit survey contained five Likert prompts that were used in an attempt to elicit participants' feelings about their experience interacting with our storytelling system. As with the demographic information, participants were not required to answer the prompts; their lack of response was recorded to enable a sensitivity analysis. We were most interested in participants' feelings of "control" over the experience and the degree to which they felt "manipulated" by the system. We were also interested in the degree to which players felt a sense of "engagement" with our system and to which they felt a "connection" with the main character. Lastly, we asked players to indicate their agreement with the perception that the story was "adapted" specifically to them. The complete set of Likert prompts from the exit survey is included for reference in Appendix B. The complete sets of responses are also included for reference in Appendix C; however, we will analyze some of these responses in this chapter below.

7.1.4 Participant Recruiting

We recruited study participants from five different locations including a special interest internet forum and mailing lists at our institution as well as two of our peer institutions. Recruiting messages asked people to help out with a "mixed-media interactive storytelling system" and the focus of the study was intentionally left vague. Further, the consent information presented to the recruits was intentionally left vague as well. It described the study as focusing on the emotional response of players to videos, even though that was not exactly our goal. We chose to do this because we did not want to bias our results by prepping participants to be on the lookout for influence messages.

Once we reached 75 participants we ceased recruiting additional participants (although the URL to our web-based system remained active). That is, of all the people who clicked on the link in our advertisement, once 75 of them had filled out (or declined to answer) the demographic survey we no longer used additional data for analysis.

7.2 Study 1 Hypotheses

As mentioned above, we conducted our first study for two main reasons: 1) to verify that influence can indeed be effective in a storytelling setting;[2] and 2) to identify any shortcomings in our study design.

Our hypotheses were:

1. We will find a significant positive change in the frequency players make the decisions we apply hand-authored scarcity statements to

2. We hypothesize there may be (although hope there won't be) a significant difference

[2]This motivation is in the spirit of experiments Reeves and Nass conducted on people's treatment of computers [99].

in players' reported levels of agreement with the following statements when influence is applied:

(a) a sense of control over the story

(b) a feeling the story was adapted to them

(c) feelings of being manipulated by the storytelling system

(d) a sense of connection with the main character

(e) a sense of engagement with the story

7.3 *Study 1 Results*

In short, the results of the study were rather encouraging, although not definitive. As the ultimate goal is to use our models of influence to shape a player's experience in a narrative environment, we opted to conduct our study in a pseudo-game environment where we were unable to control for all possible variables. Despite not controlling for everything, we can still claim with confidence that using hand-authored statements based on the principle of scarcity in an interactive storytelling environment can lead to a statistically significant increase in the frequency with which a goal specified by the author is realized compared to the frequency with which that goal is realized by a control group without explicit influence. There are various alternative explanations. The increase could be attributable to having an extra sentence, more frequent use of the word "fish", *etc*. To fully eliminate these possibilities would require an even more tightly-controlled study outside the type of narrative environment that we are interested in.

7.3.1 Data Selection

Of the 75 participants we collected data from, 72 continued to begin the actual story, and 61 completed an entire iteration of the story and answered (or declined to answer) the exit survey. Additionally, although we will not report on the data because it is beyond the scope

Table 16: The results of a χ^2 analysis and a Fisher's Exact Test on demographic survey data, exit survey data, goal realization, and participant exclusions for the first study.

Variable	χ^2	d.f.	p	Fisher p
Dropout Rate:				
excluded	0.3970	1	0.5286	0.7438
Demographics:				
education level	2.8791	7	0.8960	0.9106
age	4.8730	7	0.6755	0.7521
game hours	4.1784	5	0.5240	0.6484
internet hours	4.3739	6	0.6262	0.6791
Exit Survey:				
sense of control	9.5556	5	**0.0889**	**0.0464**
adapt	2.7028	5	0.7457	0.7798
manipulated	3.8408	5	0.5726	0.6436
connection	3.3411	5	0.6476	0.6888
engagement	3.6109	5	0.6067	0.6624
Goal Realization:				
achieve goal	5.4289	1	**0.0198**	**0.0297**

of this study and dissertation, 12 of the participants continued on to begin the story at least once more.

Of the 72 participants who began the story, we threw out the data from 11 trials where the participants did not complete the story. Because the major motivation behind this study was to determine the rate at which players entered a particular story state, including partial stories did not make sense. We were curious as to whether or not there was a significant difference between the control and treatment groups in the drop out rate. To examine this, we ran a χ^2 test and Fisher's exact test and failed to find a significant difference in the rate of study participants dropping out of the study between the groups: control ($\frac{5}{39}$) and treatment ($\frac{6}{33}$), $p = 0.5298$ (see "excluded" in Table 16).[3]

7.3.2 Summary

Table 16 summarizes statistics from χ^2 and Fisher's Exact Test analyses of participant responses. Due to our relatively small sample size and response distributions, there were a number of cases where the expected values in the χ^2 computation were less than five. In such cases, Fisher's Exact Test serves as a more appropriate statistical test that can be interpreted in the same way as a χ^2 test [38]. Accordingly, we rely largely on the p values indicated by Fisher's Exact Test.

The analysis was conducted on the data we obtained including a category for "no answer." To be sure, we re-ran the analysis on the same data set where we excluded missing responses. We found that while the p values may have changed, the statistical significance of the results at $\alpha = 0.05$ was unchanged with one exception. As will be discussed below, the "sense of control" Likert scale responses (see "control" in Table 16) were marginally significant ($p = 0.0889$) according to the χ^2 analysis but significant at $\alpha = 0.05$ according to Fisher's Exact Test ($p = 0.0464$) when no answer was included in the analysis. When no answer was excluded from the analysis, both the χ^2 ($p = 0.0490$) and Fisher's Exact Test ($p = 0.0295$) indicated significance at the $\alpha = 0.05$ level.

7.3.3 Goal Realization

The major finding of this study is that inserting influence statements based on the principle of scarcity can have a statistically significant effect on the frequency that goals are realized. Of the 61 participants who completed the story and took part in the exit survey, 34 were randomly assigned to the control group and 27 were randomly assigned to the treatment group. In the control group, 18 out of the 34 (or 52.9%) of the participants bought fish during the story (the desired goal) whereas in the treatment group, 22 out of the 27 (or 81.5%) of the participants bought fish. This difference is significant with $p = 0.0297$ (see

[3]We only counted a participant as a dropout if they clicked past the demographic survey and began with the first story event. There were three participants out of the 75 who did complete the demographic survey but did not continue to the first story event.

"achieve goal" in Table 16), indicating that the hand-authored influence statements were indeed effective.

7.3.4 Population Bias

To examine whether the significance of the goal realization result may have been biased by an unbalanced population between the treatment and control groups, we compared the distribution of answers to each of the four demographic survey questions. We did not find a significant difference in the distributions of responses for any of the four questions. In addition, to justify the exclusion of incomplete data, we compared the rate of participant dropout between the two groups. We found that five out of the 39 (12.8%) participants in the control group did not complete the story whereas six out of the 33 (18.2%) participants in the treatment group did not complete story. We did not find this difference to be significant ($p = 0.7438$).

7.3.5 Exit Survey

As we were interested in identifying any significant differences in players' responses to these measures, we ran both χ^2 tests and Fisher's exact tests on the survey responses divided by control and treatment groups. As we had hoped, we did not find a significant difference in the responses of the players in the two groups for four out of the five Likert prompts in the exit survey, specifically with the adaption, manipulated, connection, or engagement Likert prompts (indicated in Table 16). These results were unchanged in our sensitivity analysis.

We did observe a significant difference in the responses of players in the control and treatment group when asked about their feeling of control over the story progression ($p = 0.0464$). The full distribution of answers by group is presented in Figure 23. There are two points worth mentioning about the distributions. First, participants in the control group did not indicate strong feelings in either direction, tending to either agree or disagree (with only one participant selecting "neither agree or disagree"). On the other hand, some of

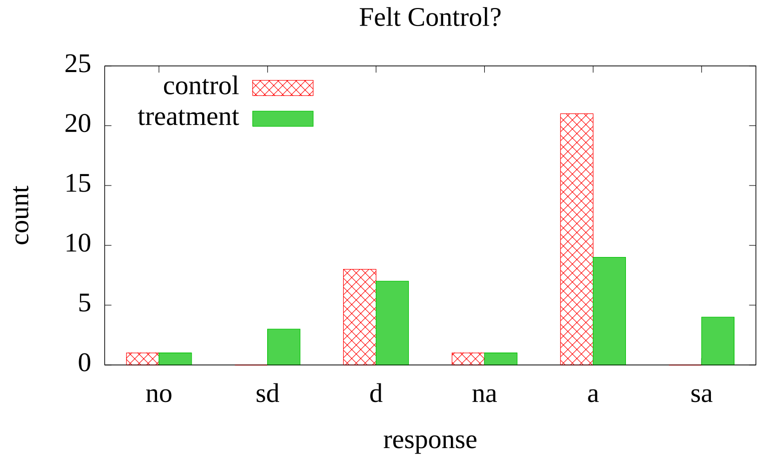


Figure 23: The distribution of responses in the exit survey during the first study pertaining to the player's perception of control over the story progression. There were 31 participants in the control group and 25 in the treatment group who responded to this prompt. The difference observed proved to be significant ($p = 0.0464$). ("no" indicates no answer given, "sd" indicates strongly disagree, "d" indicates disagree, "na" indicates neither agree or disagree, "a" indicates agree, and "sa" indicates strongly agree).

Table 17: The relative frequency of general disagreement (either "strongly disagree" or "disagree") and agreement (either "agree" or "strongly agree") with the "I felt a sense of control" Likert prompt in the first study. The "neither agree nor disagree" responses are not included in this table.

	Disagreement	Agreement
control	26.67%	70.00%
treatment	41.67%	54.17%

the participants in the treatment group indicated a stronger level of agreement with four responses in the strongly agree category and a stronger level of disagreement with three responses in the strongly disagree category. It is unclear why we observed more extremal responses from participants in the treatment group.

Second, note in Figure 23 the discrepancy between participants in the control and treatment groups who indicated overall agreement (either agree or strongly agree) or overall disagreement (either disagree or strongly disagree). We present this data in Table 17. In the treatment group, 13 out of 24 (54.17%) indicated overall agreement whereas 21 out of 30 (70.00%) in the control group indicated overall agreement. On the other hand, we observed a higher percentage of treatment group participants indicating overall disagreement when compared to control group participants. In the control group eight out of 30 (26.67%) indicated overall disagreement whereas 10 out of 24 (41.67%) did so in the treatment group.

7.4 Study 1 Discussion

Here we discuss the findings of the first study. Our first hypothesis for this study pertained to the use of influence to effect a significant change in player decisions during the story. The data presented above describing the goal realization rate for players in the two groups supports this hypothesis: using scarcity statements can significantly increase the rate at which players choose to realize a goal to which influence has been applied. Our study design does not afford the ability for us to characterize if it is influence itself, or an artifact of the presentation of influence (*e.g.*, more words, a specific word, etc.) that was the cause for this effect; however, that specific cause isn't important if we know that influence works.

We hypothesized that there might be a significant difference in participants' stated agreement with our five Likert prompts, although we hoped there would not be. The responses to four of the five prompts did not indicate a significant difference. We did not find a significant difference between the control and treatment group for the manipulated, adapted, connection, and engagement prompts. This was as we had hoped; however, we did find a significant difference in players' responses to the control Likert prompt.

We were surprised by the significant finding in the responses to the sense of control Likert prompt. One explanation is that the control group in this study is equivalent to no drama manager. Thus, it makes some sense that participants perceived a change in their degree of control in the treatment group. An alternative explanation which we believe is the more probable one revolves around story content (data from the second study presented below in Section 7.7.2 lends credence to this explanation as well). Players who did not purchase fish entered a portion of the story where there were more options immediately available to them. While the branching factor of the story graph is two for all paths, there was more immediacy between events in the paths that do not pass through the "purchase

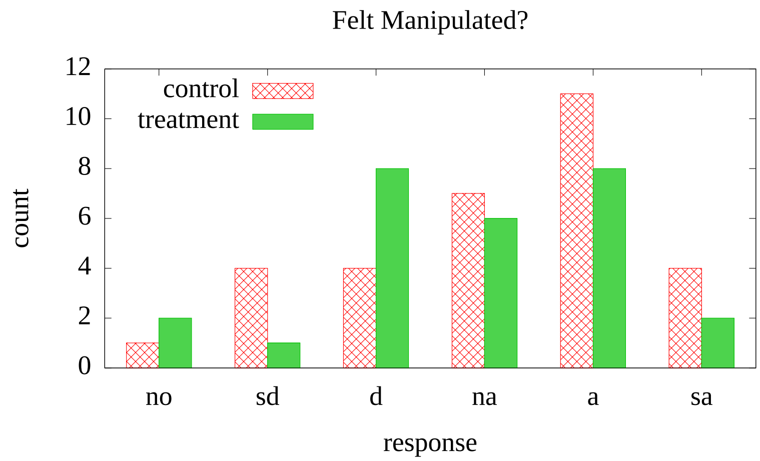


Figure 24: The distribution of responses in the exit survey during the first study pertaining to the player's perception of being manipulated. There were 31 participants in the control group and 27 in the treatment group who responded to this prompt. The difference observed did not indicate significance ($p = 0.6436$). ("no" indicates no answer given, "sd" indicates strongly disagree, "d" indicates disagree, "na" indicates neither agree or disagree, "a" indicates agree, and "sa" indicates strongly agree).

fish" (the goal) event; that is, if a player made a decision to cause a particular story event, it happened immediately and they were presented with another set of options. The path that led from purchasing fish required, on multiple occasions, for the player to re-commit to a decision they had made earlier. For example, the first time the player is given the choice to purchase fish, the next story event puts them in a situation where they pass the train station and must decide whether or not to continue with their original plan to buy fish or to enter the train station and head out of the city for the day. Anecdotally, a number of participants described this as "frustrating." Because we did not include an interview or open-response question in our survey, we cannot report any qualitative results to back up this explanation. Further, this characteristic of the story only became apparent to us in a *post hoc* analysis after speaking informally with a handful of study participants.

Given this anecdotal evidence and alternative explanation, an additional point of interest is the apparent lack of significant difference in a player's feelings of being manipulated (as illustrated by the p value in the "manipulated" line of Table 16). The distributions of

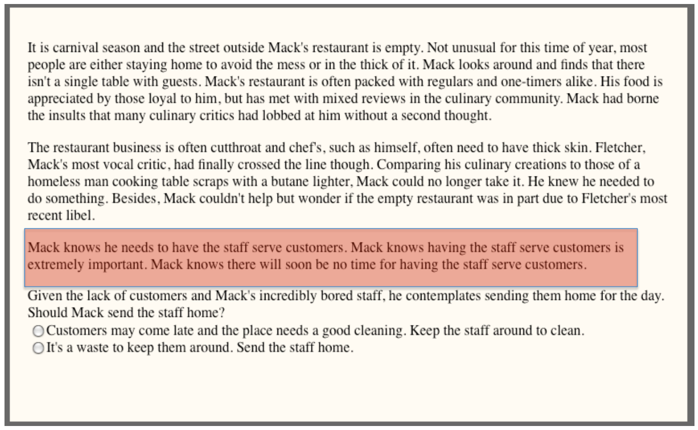


Figure 25: A screenshot of the storytelling system during a story event from the second study. The player sees text containing story information and a question-response set that allows the player to make choices for the main character. The area highlighted for emphasis illustrates an influence statement employed by the system.

answers by group for feeling manipulated are presented in Figure 24. These data are very encouraging when taken in concert with the alternative explanation. If it is in fact the case that participants did not feel any more manipulated when influence statements were used and the change in feelings of control are attributable to story structure rather than influence, then we likely have an indication that the player's sense of self-agency is preserved when models of influence are used. The results presented in Section 7.7.2 lend further support to this explanation.

7.5 Study 2 Methodology

Armed with the results of our first study and these goals, we designed our end-to-end evaluation. In this section, we will describe the story used and how it was authored based on the results of the first study. Additionally, we will describe the implementation of each of the DODM components for this story environment. Here we focus on methodology for study 2 that is different from study 1—anything not explicitly discussed in this section can safely be assumed to have been implemented as it was for study 1.

Unlike in the first study, we opted not to use videos in the second study. Figure 25 contains a screenshot of the study interface during a story event from the second study.

The important thing to note is the lack of a video. We chose to leave videos out of the second study for two reasons: 1) After having spent far more time searching youtube for appropriate videos than writing the story used for the first study, we decided the effort wasn't necessary; and 2) as the videos don't enable us to measure anything relevant to our hypotheses, there is no need to include them. Outside of this difference, the architecture for the second study was identical to the first.

7.5.1 The Story

One of the confounding results we obtained from the first study was the different levels of agreement with the "sense of control" Likert prompt between the control and treatment groups (see Section 7.3.5). We had hypothesized that the likely cause for this result was not the use of influence, but the story content that resulted from the successful application of influence. To control for that in our second study, we ensured that the story structure was essentially "symmetric." To implement this symmetry, we based our narrative on a linear story—"The Cask of Amontillado" by Edgar Allan Poe.[4] To convert this linear plot line into a branching story structure for our purposes, each event in the linear plot was considered a "level" in the tree. Thus, the second event in Poe's story was one of two possible events in our branching story. Similarly, the third event in Poe's story was one of four possible events in our branching story. *Etc.* Our story had a minimum depth of five events and a maximum depth of seven events. There were a total of 22 story events, with six different questions, 12 possible answers, and 32 possible transitions.

Our story was basically Poe's story, although the names of characters and the setting was modernized. For example, Poe's main character was named Montressor, and our main character's name was Mack. Montressor's antagonist was Fortunato and Mack's was Fletcher. Rather than focusing on a rare cask of amontillado in a wine cellar, our story was centered around a rare side of kobe beef in a restaurant freezer. The parallels between Poe's

[4]"The Cask of Amontillado" first appeared in *Godey's Lady's Book*, vol. XXXIII (November 1846).

story and ours were essentially a one-to-one mapping. We chose to keep the similarities strict as a means to further control for the story content's effect on players' experiences. Poe's story is very popular and hailed by some as his finest short story in the horror genre. By leveraging his story structure, we hoped to limit the effect on a player's engagement we might encounter as a result of a poorly authored story. The deviations from Poe's story that we authored were all in the endings. There were 34 possible stories, or trajectories through our branching story graph, which corresponded to five possible endings—only one of which was the original ending of Poe's.

7.5.2 The Influence Treatment

Having evaluated the effects of hand-authored influence using the first study, we opted to relax our control over the variables in the second study slightly. One of the ways in which this manifested itself is in the implementation of influence for the treatment condition. Whereas before pre-authored influence of only one type was applied to realize one particular story event, in the second study we authored three schemata for each of two different types of influence and made at least one of them available to the system to apply to every possible decision the player was faced with (should it choose to apply it that often). The influence types used were scarcity and reciprocity. We opted to use scarcity in part to compare the effectiveness of our generated influence statements to the pre-authored scarcity statements used in the first study. On the other hand, we chose to use reciprocity because it has vastly different characteristics to scarcity in terms of implementation: it requires a give-and-take question-response sequence. The complete set of six influence schemata for scarcity and reciprocity used in our study are included for reference in Section F.2 of Appendix F.

For both types of influence, we strictly adhered to the "base story + influence" paradigm in an effort to facilitate data analysis. In the case of scarcity, this is a natural fit as it is implemented in a one-way communication between the system and the player. In the case of reciprocity, a possibly more natural (and maybe even useful) way to implement it would be to replace the story's question with the reciprocity question as the trigger for the next story event; in order to keep the paradigm consistent and avoid the introduction of an additional confounding factor to our results, we chose to decouple the reciprocity question from the story question and, in part, rely on the player to remain consistent with their answer to the reciprocity question when answering the story question. In Section 7.7.3 we will discuss the degree to which this was effective or not.

7.5.3 The Locus of Control Scale

Due to the significant findings in the first study pertaining to the sense of control Likert prompt, we chose to add an additional section to the exit survey. The answers to the questions in that section may help to disambiguate another unexpected significant finding in the sense of control Likert prompt for the second study. *Locus of Control* (LOC) is a term from psychology that describes the degree of belief someone has that the good or bad things in their life are the result of their own actions/decisions or result from some external forces beyond their control. Those who have a high locus of control score tend to believe that external forces determine the events in their life whereas those with a low score tend to believe they are in control of their own destiny. The scale was developed by Rotter [116] and has become a staple of personality studies in social psychology. We seek to leverage Rotter's work on locus of control to further describe how the players of our interactive story perceive a sense of control—and by extension self-agency—over the experience.

The original LOC scale consists of 23 forced-choice alternative questions plus six filler items. Some of the original questions pertained to generic events in people's lives such as "unhappy things" or "misfortunes." The other types of questions in the original scale pertained to external events such as "wars" or "teachers being unfair to students." In order to adapt the LOC scale questions for use in our system, we excluded the six filler questions and identified the subset of the 23 questions that were generic enough to be adapted to pertain to our story. For example, a question that referred to "people's misfortunes" was rewritten to say “Mack’s misfortunes” (where Mack was the name of the character our study participants played). Of the 23 question original questions, we eliminated 10 of them, resulting in the 13 question LOC scale that was given to study participants as part of the exit survey during our end-to-end evaluation. This scale results in a score between zero and 13. As with the Likert prompts, participants were not required to answer the questions; their lack of response was recorded to enable a sensitivity analysis. Those questions are included for reference in Appendix B. Note that we have annotated the presentation of those questions with the scoring key, but those annotations were not visible to our study participants.

7.5.4 Participant Recruiting

To recruit participants for our study, we advertised using personal emails and messages to mailing lists, posts to special interest internet forums, social networking sites, and class distribution lists (email or otherwise). Our IRB protocol was approved for far more participants than we expected to be able to recruit, so in our recruiting messages we encouraged people to recruit and share the study with their friends as well. Since the number of participants was not the limiting factor in our recruiting, we collected data for a three week period. At the end of that period, the response rate had dropped to near zero per day, but the study remained open for additional users should they choose to participate; however, any data collected beyond the three week period was not included for analysis. Like the first study, the recruiting messages were intentionally left vague, mentioning only an “interactive storytelling” project so as not to bias results by priming participants with influence concepts.

7.5.5 The DODM Components

In order to implement a TTD-SMDP for our study, we needed to implement each of the components: trajectories, drama manager actions, a player model, and a target distribution. Here we describe each of those components as implemented for our evaluation.

7.5.5.1 Trajectories

The branching story graph we constructed (and described briefly above) sits upon the set of 22 story events. These events have precedence constraints induced by the fully-connected directed acyclic structure of our graph. That is, the first event is the only event that could have occurred at the beginning of the story. Then, the only available events after the first are those at level two of the graph. Once one of those occurs, then the only available events are those at level three of the graph that immediately succeed the level two event that has just occurred. *Etc.* Thus, paths from the source of the story graph to the sinks comprise the trajectories for the TTD-SMDP.

7.5.5.2 Drama Manager Actions

The actions for our study were the influence schemata (see Section F.2 of Appendix F for reference). This is in contrast to the first study where influence was hand-authored. We implemented six different schemata, three for the scarcity rule of influence and three for the reciprocity rule of influence. Note that not all of the actions were applicable in every trajectory. For these six schemata, we required 53 separate templates (some with multiple conjugations) in order to ensure that unification would succeed. Note that we have not included these templates in this dissertation. We arrived at 53 templates through a process of examining the templates required by the schemata (*e.g.*, an action in a schema `speaks(<npc>,<player>,a-gift)` requires a “speaks” template and an “a-gift” template).

Recall from the description of DODM in Chapter 1 and Chapter 2 that actions “operate on plot events.” In our setting, the forced-choice question that drives the narrative forward links an answer with a plot event deterministically. In effect, our actions operate on answer choices as a means to operate on plot events. Thus, in addition to the templates required by each of the schema, for every answer (12 of them) a set of “input templates” had to be authored for each schema that we wanted to be available to apply to that answer. We ensured that every possible answer had at least one valid set of input templates associated with it. Where not overly burdensome or when it made sense to do so, we authored input templates for more than one schema to allow our algorithms to make the choice about which to apply.

It is worth noting that the authorial effort of creating these 53 templates is not an accurate representation of what may be required for using these techniques in the future. The set of templates we authored covered both the templates "hard-coded" in the schema actions as well as the set of templates supplied as input to the unification process. The set of templates coded in the influence schemata are *reusable* across domains. Thus, to implement these methods in another story domain, the burden on the author would be to supply only those templates used as input for unification, a job requiring a significantly reduced effort in comparison to that we put forth.

7.5.5.3 Player Model

The vast majority of work on the DODM formalism and its associated player transition model has assumed the player will behave uniformly randomly. The notable exception is the EMPath system [23, 121, 122] which based its player model on actual physical distance between the player and the next plot event. Other than distance, however, there is no assumption made about how players will behave and a similar prior of uniform behavior is assumed. We continue that convention here as well; however, because we use a text-based experience, physical distance plays no role in player behavior. Thus, our player transition model is assumed to be uniformly random when no drama manager action has been applied. To author the conditional probabilities that describe player behavior when actions are active, we followed the procedure for transferring probabilistic data across domains presented in Section 5.5. See the "goal_prob" field in the six influence schemata in Section F.2 of Appendix F for the specific calculation results as well as references to the input data.

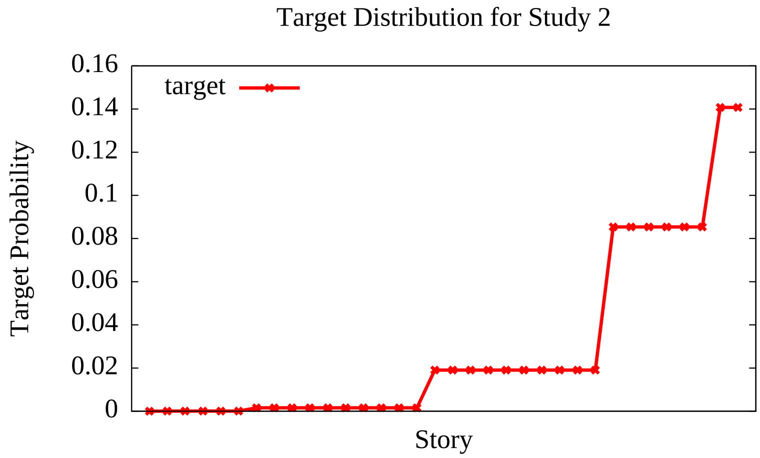


Figure 26: The target distribution for the second study. The x-axis is an arbitrary indexing over complete stories sorted in ascending order of target probability. The y-axis indicates the target probability.

7.5.5.4 Target Distribution

After a fairly thorough analysis of the differing performance characteristics of the various authoring paradigms for target distributions of TTD-MDPs (Chapter 6), we opted to create the target distribution for our study using the prototype-distance paradigm using hand-selected prototypes (as opposed to sampled). There were two main motivations for this: First, as we are not narratologists we felt that an evaluation function that we might author for use in other paradigms would be contrived at best; and second, we are hopeful that the prototype-distance paradigm is the one that authors will eventually find most intuitive and useful, when we are able to evaluate it with actual authors. This will be discussed further in Section 8.4.1.

For our study, we chose two prototypes out of the 34 possible stories. They were chosen, in part, based on the original story structure of Poe's version of our story, and because they resulted in different endings that we hoped players would experience. We used Levenshtein distance [62] as our distance measure, and used a variance of 1.0 for our Gaussian models. The result of our target distribution is presented in Figure 26.

7.6 *Study 2 Hypotheses*

There were two main goals in conducting the end-to-end evaluation of our system. The first was to verify that all of the theoretical connections between our techniques are realizable in practice. The second was to show that when those connections are made, that a DODM drama manager using TTD-SMDPs, influence schemata, and natural language templates is effective at shaping players' experiences according to the goals specified by authors, and is able to do so without a player perceiving a decrease in their sense of self-agency. A third lower-priority, but still important, goal was to further examine the effectiveness of using influence as a tool to shape player decisions and to characterize the performance of two types of influence: scarcity and reciprocity.

Our hypotheses were:

1. We hypothesize there may be (although hope there won't be) a significant difference in players' reported levels of agreement with the following statements when influence is applied:
   (a) a sense of control over the story
   (b) a feeling the story was adapted to them
   (c) feelings of being manipulated by the storytelling system
   (d) a sense of connection with the main character
   (e) a sense of engagement with the story
2. We hypothesize there may be (although hope there won't be) a significant difference in the means of Locus of Control scores for the participants in the control and treatment groups
3. We will find a significant positive change in the frequency players make the decisions we apply system generated and refined scarcity and reciprocity statements to

4. The use of DODM with TTD-SMDPs, influence schema, and natural language templates in an interactive story will result in a distribution of stories closer (as measured by KL-divergence) to a target distribution than when those techniques are not applied

7.7 Study 2 Results

In this section, we will present the findings of our end-to-end evaluation. For the most part, our findings were consistent with the findings of the first study. They confirmed the key conclusion of the first study that implementations based on at least one method of influence from social psychology, the scarcity method, have a significant effect on the choices players make; however, the results also led to some interesting surprises as not every type of influence we implemented yielded the expected significant results. The complete distribution of responses to the demographic survey and exit survey are included for reference in Appendix D. Here, we focus on the presentation of summary statistics and important findings.

7.7.1 Data Selection

Of all the people who clicked on the link in our advertisements during the three week recruitment period, 206 of them clicked past the consent information and either filled out or declined to answer the demographic survey questions. Of those 206 participants, 191 of them continued past the instructions and began the story, and 166 of them completed an entire episode of the story. 77 participants continued to begin a second episode of the story and 39 completed an entire second iteration. Only eight participants continued beyond a complete second iteration of the story.

In the analysis we present below, we have used the same exclusion criteria used in our first study: if the participant did not complete an entire episode of the story, we excluded their data. Further, if they chose not to complete the exit survey but did complete the story, we included their story trace for those analyses but excluded their data for the analyses of the exit survey. A sensitivity analysis has indicated that significance findings discussed here

Table 18: The results of a χ^2 analysis and a Fisher's Exact Test on demographic survey data, exit survey data, and participant exclusions for the second study.

Variable	χ^2	d.f.	p	Fisher p
Dropout Rate:				
excluded	0.11	1	0.7401	0.8597
Demographics:				
education level	5.9149	7	0.5497	0.5574
age	8.1709	8	0.4170	0.4157
game hours	5.5997	6	0.4695	0.5640
internet hours	2.3369	6	0.8863	0.9300
Exit Survey:				
sense of control	3.1399	5	0.6784	0.7416
adapt	5.2789	5	0.3828	0.3716
manipulated	4.8975	5	0.4285	0.4666
connection	9.0875	5	**0.1056**	**0.1108**
engagement	8.6105	5	**0.1256**	**0.1263**

did not change if the excluded data were included in the analyses in the following ways: partial story information was included when examining the effectiveness of influence and a lack of response to an exit survey question was included in a new "no response" category.

Of the 206 participants who clicked past the demographic survey, 113 of them were randomly assigned to the control condition and 93 of them were assigned to the treatment condition. 92 of those participants assigned to the control group completed the story whereas 74 of those assigned to the treatment group completed the story. Thus, the "dropout" rate (the participants we excluded) among the control group was $\frac{21}{113} = 18.6\%$ and among the treatment group was $\frac{19}{93} = 20.4\%$. This difference did not appear significant ($p = 0.8597$ according to Fisher's exact test).

7.7.2 Summary

Here we briefly discuss the results of the demographic and exit surveys. Table 18 contains the summary statistics for all of the demographic survey responses and exit Likert prompts. It is structured similarly to Table 16 above where the results of the first study are presented.

The significant finding in the first study for the "sense of control" Likert prompt was not

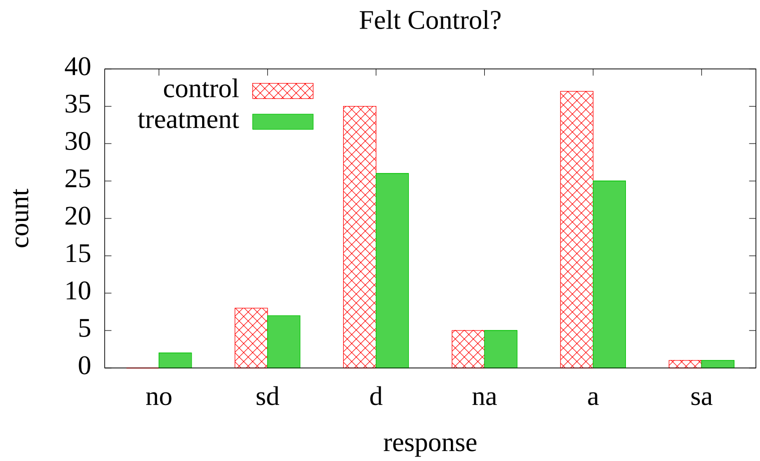


Figure 27: The distribution of responses in the second study for the Likert prompt: "I felt a sense of control over the story progression." There were 86 participants in the control group and 66 in the treatment group who responded to this prompt. The difference observed did not indicate significance ($p = 0.7416$). ("no" indicates no answer given, "sd" indicates strongly disagree, "d" indicates disagree, "na" indicates neither agree or disagree, "a" indicates agree, and "sa" indicates strongly agree).

Table 19: Table of summary statistics for distribution of Locus of Control scores.

	mean	*std dev*
control	6.627907	2.841071
treatment	6.272727	3.05108

found again. We had hypothesized that story content may have effected player responses during the first study. While we cannot be sure that was the case, the lack of significant finding in the second study is as we had hoped. Figure 27 contains the distribution of responses to that Likert prompt. Note that these data are reported as counts and that there were more participants in the control than treatment group. Thus, the slightly higher levels in the "disagree" and "agree" categories for the control group are negligible when sample size is adjusted for.

To further characterize feelings of control, we examined the participants' modified Locus of Control scale scores. A plot of those scores is shown in Figure 28. To understand these responses, we looked at central tendency and dispersion of the distribution of the

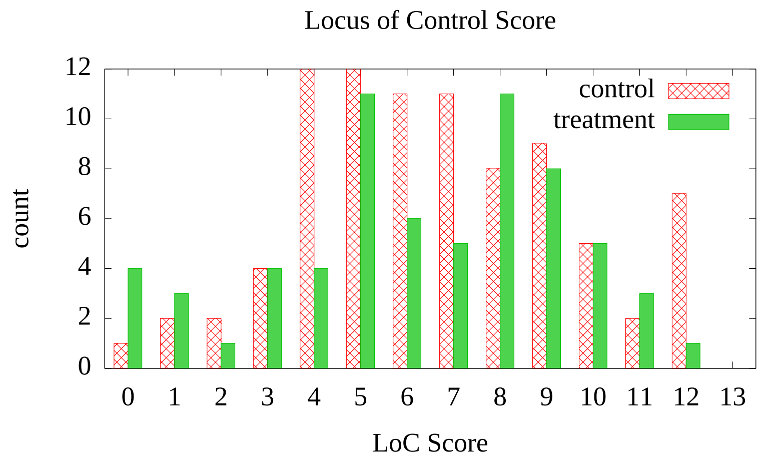


Figure 28: The distribution of Locus of Control scores in the second study. There were 75 participants in the control group and 66 in the treatment group who answered all 13 questions pertaining to this scale.

scores. Table 19 contains those summary statistics. Of interest in that table is the essentially equal means and standard deviations for the distribution of scores for the participants in both conditions. We performed a two sample t-test and failed to find a significant difference between the means of the LOC scores for the two groups ($p = 0.4649$).

The last few results from the end-to-end evaluation we would like to discuss are the players' responses to the "connection" and "engagement" Likert prompts. The full distributions of those responses are presented in Figure 29 and Figure 30. In Table 18 above, the entries in bold-faced font correspond to the p values associated with the answer distributions for these two Likert prompts. While these values do not meet our significance criterion of $\alpha = 0.05$, they are quite a bit lower than any of the other p values.[5] We will discuss this finding further in Section 7.8.3 below.

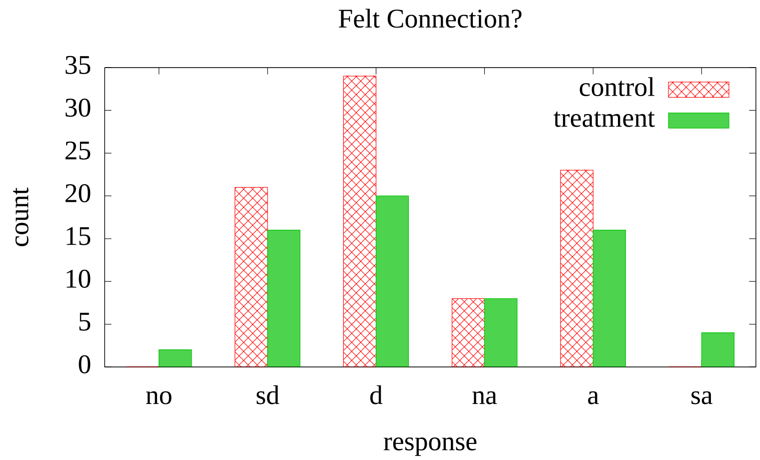


Figure 29: The distribution of responses in the end-to-end evaluation for the Likert prompt: "I felt a sense of connection with the character in the story." There were 86 participants in the control group and 66 in the treatment group who responded to this prompt. The difference observed indicated marginal significance ($p = 0.1108$). ("no" indicates no answer given, "sd" indicates strongly disagree, "d" indicates disagree, "na" indicates neither agree or disagree, "a" indicates agree, and "sa" indicates strongly agree).

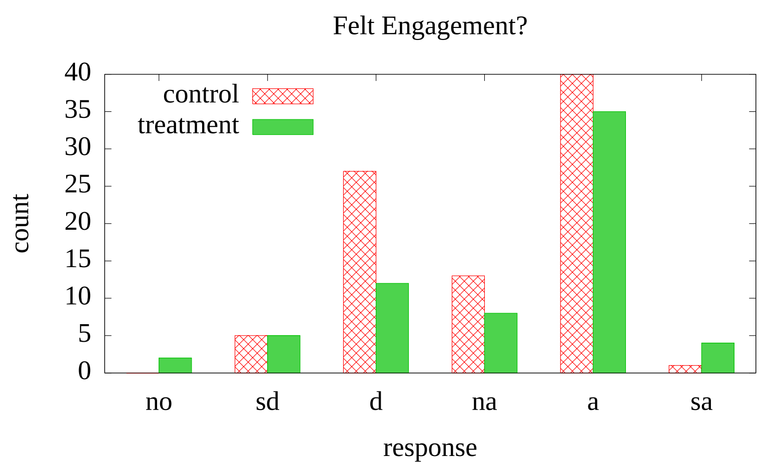


Figure 30: The distribution of responses in the end-to-end evaluation for the Likert prompt: "I felt a sense of engagement with the system." There were 86 participants in the control group and 66 in the treatment group who responded to this prompt. The difference observed indicated marginal significance ($p = 0.1263$). ("no" indicates no answer given, "sd" indicates strongly disagree, "d" indicates disagree, "na" indicates neither agree or disagree, "a" indicates agree, and "sa" indicates strongly agree).

Table 20: The effectiveness of influence grouped by story depth. The table entry in either the "a_1 prob(count)" or "a_2 prob(count)" column annotated with an "(*)" indicates the answer choice influence was applied to. The p value reported here is from Fisher's exact test.

Depth		a_1 prob(count)	a_2 prob(count)	p value	**Type**
1:	control	0.7065(65)	0.2935(27)		
	treatment	(*)0.8514(63)	0.1486(11)	**0.0401**	scarcity
2:	control	0.2283(21)	0.7717(71)		
	treatment	(*)0.2568(19)	0.7432(55)	0.7170	reciprocity
3:	control	0.1630(15)	0.8369(77)		
	treatment	0.2297(17)	(*)0.7703(57)	0.3246	scarcity
4:	control	0.7609(70)	0.2391(22)		
	treatment	0.7703(57)	(*)0.2297(17)	1.0000	reciprocity
5:	control	0.5152(17)	0.4848(16)		
	treatment	(*)0.2500(2)	0.7500(6)	0.2488	reciprocity
	treatment	0.3889(7)	(*)0.6111(11)	0.5580	scarcity
6:	control	0.7500(3)	0.2500(1)		
	treatment	(*)1.0000(2)	0.0000(0)	1.0000	reciprocity

7.7.3 Influence Effectiveness

An important consideration in our data analysis was to characterize how effective influence was. To illustrate this, we needed to compare distributions of answer choices across the populations. The data resulting from this analysis is presented in Table 20. Here we will explain these findings in detail and offer some potential explanations. In short, we found that reciprocity did not affect players' decisions and that scarcity was effective in certain situations and not in others.

To start, consider the organization of the table. Recall from the description of the story in Section 7.5.1 that it was organized into depths. Every event at a particular depth had the same question to move the narrative forward. Thus, to better evaluate the data, we grouped the effectiveness of the influence techniques by depth rather than by event. The "depth" column of the table indicates this. The next two columns of the table contain the probability

[5]While preparing our data analyses we examined results using the data collected during the first week of our three week data collection period. The p values associated with the "connection" and "engagement" Likert prompts, while not an indicator of significance in the ultimate results, did indicate significance before data collection was completed.

with which each answer was selected by the player and the raw count in parenthesis. The answer column annotated with an "(*)" indicates the answer that the influence technique in the treatment condition had targeted. The fifth column contains the p values which indicate the statistical significance of the alternatives in the table, using Fisher's exact test. Finally, the last column indicates the type of influence used.

There are a number of interesting things to note. First, there is a statistically significant effect of influence—namely the result of the application of scarcity in the first story event. This confirms our observation in the first study. It was at first surprising that this was the only statistically significant effect we found. At face value, this may appear to be a negative result—and to some degree it is; however, there are a few factors to consider. For example, consider the other uses of scarcity at depth three and five. At depth three, the baseline transition model (when no influence was applied) resulted in $P(a_1) = 0.1630$ and $P(a_2) = 0.8369$. Applying influence to a_2, as the drama manager chose to do, and getting a statistically significant result would require a notably larger number of study participants—the significance of smaller increases occurs when larger numbers of participants are used. In the third instance of scarcity use (at depth 5), we again failed to see any statistical significance. In this case, however, the baseline distribution is roughly uniform and the resulting shift after scarcity was applied was in the expected direction (*i.e.*, the answer to which scarcity was applied was selected by players more frequently). Unfortunately, because the vast majority of participants took a path through the story that ended at fifth story event (and therefore didn't get any influence applied at that level), and further because approximately $\frac{1}{3}$ of those participants who continued past depth five were steered in the other direction, the magnitude of the effect we observed for scarcity was too small for statistical analysis to make sense. Taken together, all of these indicators are encouraging for the use of scarcity in interactive storytelling environments.

The next result we wish to highlight is that reciprocity seems to not have been effective. In none of the four depths at which it was applied did it yield a statistically significant

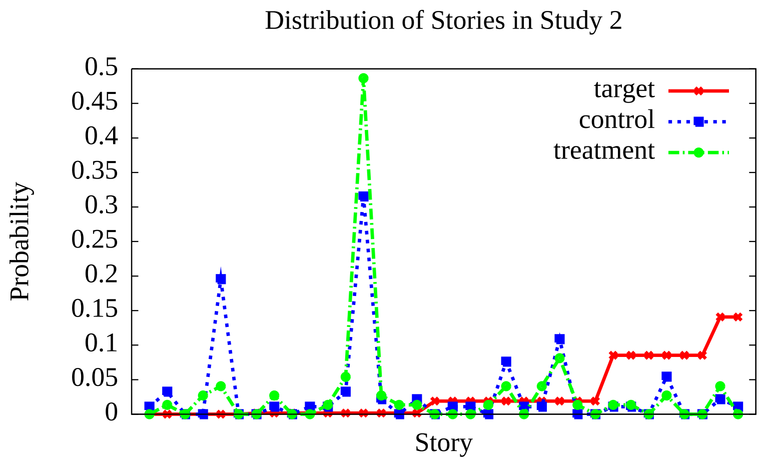


Figure 31: The full distributions of stories for the second study. The x-axis is an arbitrary indexing over complete stories sorted in ascending order of target probability. The y-axis indicates the probability (target or observed) that a story occurred or was targeted (in the case of the target distribution). Included in this plot is the target distribution, the distribution of stories experienced by participants in the control group, and the distribution of stories experienced by participants in the treatment group.

change in player decisions. This is an interesting and surprising finding, but one that we believe provides valuable insight into the applicability and design of influence models for storytelling environments. To some degree, the lack of significance is a result of a shortage of data (fewer participants experienced the longer stories than we had hoped). Specifically, the number of participants who experienced reciprocity at depths five and six was so few that we could not attribute statistical meaning to our data. On the other hand, we did find that the results at depths two and four had sufficiently large sample size to show a potentially significant effect if one existed; however, by the choice of the drama manager reciprocity was applied to low probability alternatives so its effect may have been greatly reduced. Further, what was interesting was that at depths two and four where there were a sufficient number of samples the response rates were essentially unchanged by the application of reciprocity. We will discuss this further in Section 7.8.2.

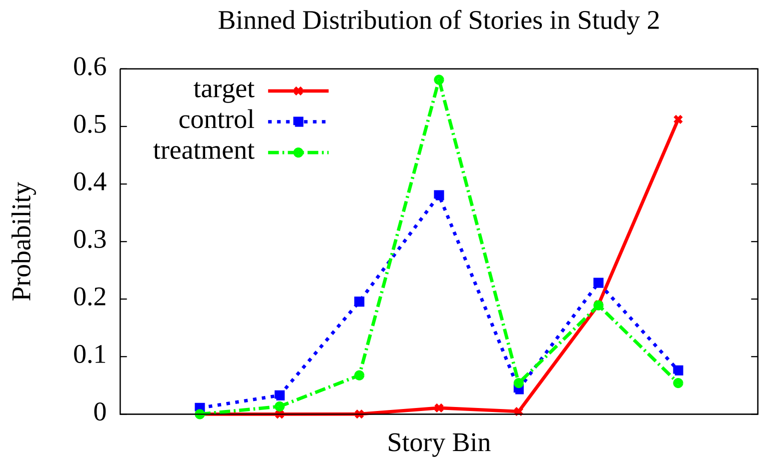


Figure 32: The full distributions of stories for the second study with "cumulative binning." The x-axis is an arbitrary indexing over complete stories sorted in ascending order of target probability. The y-axis indicates the probability (target or observed) a story in that bin occurred during an episode or was targeted (in the case of the target distribution). Included in this plot is the target distribution, the distribution of stories experienced by participants in the control group, and the distribution of stories experienced by participants in the treatment group.

7.7.4 Target Distribution Matching

Lastly we look at the distribution of stories. The primary goal of composing DODM with TTD-SMDPs, influence schemata, and natural language templates is to effect a distribution of stories that more closely matches the target distribution than a distribution of stories that occurs when our methods are not used. These results will look similar to those used in the presentation of the simulations results (Chapter 6). Figure 31 contains three curves: the target distribution, the distribution of stories for the control group, and the distribution for the treatment group. The x-axis of the figure represents an arbitrary story index, sorted so that the target probability (y-axis) of ascending indices would be monotonically non-decreasing. This means that a story that is indexed farther to the right of the plot had a higher target probability. Figure 31 is somewhat difficult to read, especially at a glance. To make this somewhat more readable, Figure 32 contains the same data projected into a lower dimensional space. Here, the x-axis is still an index indicative of the target probability; however, all of the stories with the same target probability have been binned and their cumulative (probability or frequency) mass has been plotted on the y-axis. While some information is lost by binning these frequencies, we feel this graph is important to illustrate the point: in five out of the seven histogram bins the treatment distribution is nearer (or approximately equal in distance) to the target distribution than the control distribution. This is the type of result we were hoping for.

To make this more precise, recall from Section 3.3.3 that the goal of the DM policy built on a TTD-SMDP is to minimize the KL-divergence between the target distribution and the distribution realized by combining the policy with the transition dynamics (in this case the use of influence schemata with the player's answers). Therefore, the "closer" the treatment curve in Figure 31 is to the target curve, especially in relation to the control curve, the better. Statistically speaking, $D_{KL}(target||control) = 0.96579$ and $D_{KL}(target||treatment) = 0.51029$. Thus, the DODM drama manager using a TTD-SMDP solution technique was successful at reducing the KL-divergence which is an indicator that our entire system works form end-to-end as designed to do. Note that KL-divergence is unitless, so it is not possible to characterize the significance of this result with any confidence.

One thing we would like to point out is that despite being somewhat closer to the target distribution, neither the treatment nor control distribution are particularly close to it. The only thing that might make the control distribution closer to the target distribution is a change to the story—the story itself is the way in which players in the control group decisions are influenced. Thus, the effective application of influence actions by the drama manager is mediated by the distribution of player responses in the control group. The farther the control group's story distribution is from the target distribution, the harder it will be for the application of influence to shift the distribution of stories close to the target distribution. Further, as will be discussed in the next section, not every implementation of influence we furnished to the DM was effective. These two reasons combined help to explain why the treatment and target distributions don't appear more similar.

Additional KL-divergence statistics and graphs for increasing numbers of episodes are included for reference in Appendix D.

7.8 Study 2 Discussion

Here we discuss the findings of the second study. As before, we will begin by examining our supported hypotheses. Then we will discuss our unsupported hypothesis pertaining to the use of reciprocity. Additionally, we will comment further on the responses to the engagement and connection Likert prompts, that while technically supported our hypotheses provide some interesting insight. We will finish this section with conclusions from the second study.

7.8.1 TTD-SMDPs, Influence Schema, NLG Templates are Effective

We hypothesized that there might be a significant difference in participants' stated agreement with our five Likert prompts, although we hoped there would not be. The responses to all five of the prompts did not indicate a significant difference. Further, we hypothesis there might be a difference in the means of Locus of Control scores between the groups, although we hoped there would not be. The results of the two sample t-test indicate there is no reason to believe a significant difference between the control and treatment groups exists. Taken together, we feel confident that players do not perceive any change in their sense of control over the story experience when influence is applied, and as a result should not perceive any change in their sense of self-agency either (at least when story content is controlled for).

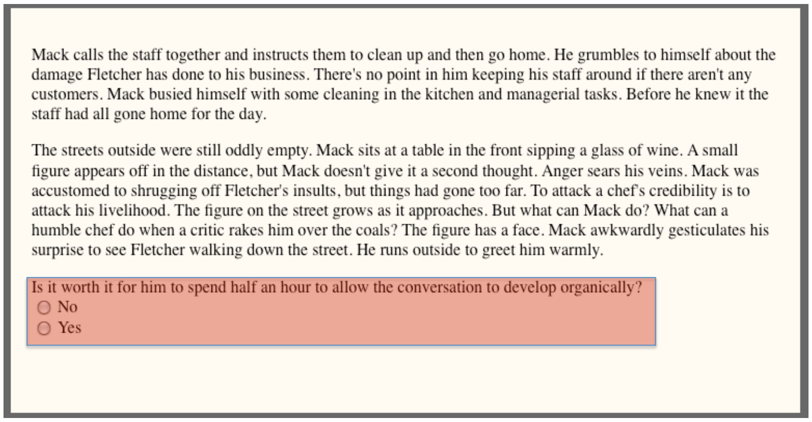
Our third hypothesis, that system-generated scarcity and reciprocity will effect significant change in player response, was partially supported. Although scarcity did not result in a significant effect in both situations where it was implemented by the drama manager, we confirmed the results obtained in the first study indicating that scarcity statements can effect a significant change in player decisions. In the situation where scarcity failed to produce a significant effect, we believe the results were confounded by a lack of data. On the other hand, we found reciprocity to be ineffective and have devoted a more in-depth discussion to this topic below.

Our fourth hypothesis for the second study, that our techniques applied to an interactive story will more closely match a target distribution, was also supported. Both a visual inspection of the binned histogram and computation of the KL divergence summary statistics indicated that our end-to-end approach using DODM, TTD-SMDPs, influence schema, and natural language templates resulted in a distribution of stories realized by players that was closer to the target distribution than when those techniques were not used.

7.8.2 The Failure of Reciprocity

To try to further uncover why reciprocity did not prove effective, we observed six participants play through the first two story events in the treatment condition and then asked them about their experiences. Note that no trace or survey data was collected from these participants and therefore their responses are not reflected in the data presented above. Of those six, four were familiar with our techniques and two were completely unaware. As in the actual evaluation, scarcity was used for the first story event and reciprocity was used for the second story event. What we found was quite interesting: the failure of reciprocity was likely due to a failure of our implementation of it in two ways, structurally and psychologically. To understand these qualitative findings, we first discuss how scarcity was implemented.

Consider Figure 25 from above where a screenshot of the evaluation study in the treatment condition with a scarcity schema active is shown. The region of the image that is emphasized shows the scarcity statement as an additional bit of text that is added to the story event. Now consider Figure 33(a) and Figure 33(b) where a story event in which a reciprocity schema was used is depicted. In Figure 33(a), the “first action” of the schema is emphasized. This action asked the player if they were willing to commit “half an hour of Mack’s time” to accomplish a particular goal. If they answered “yes” to that question,

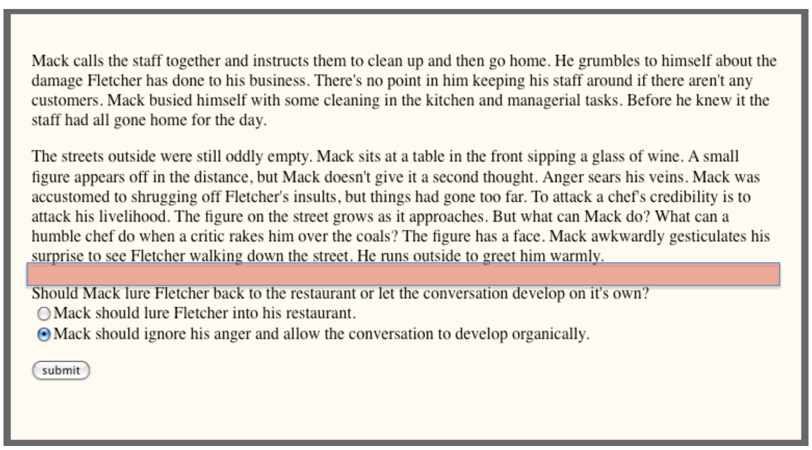


Mack calls the staff together and instructs them to clean up and then go home. He grumbles to himself about the damage Fletcher has done to his business. There's no point in him keeping his staff around if there aren't any customers. Mack busied himself with some cleaning in the kitchen and managerial tasks. Before he knew it the staff had all gone home for the day.

The streets outside were still oddly empty. Mack sits at a table in the front sipping a glass of wine. A small figure appears off in the distance, but Mack doesn't give it a second thought. Anger sears his veins. Mack was accustomed to shrugging off Fletcher's insults, but things had gone too far. To attack a chef's credibility is to attack his livelihood. The figure on the street grows as it approaches. But what can Mack do? What can a humble chef do when a critic rakes him over the coals? The figure has a face. Mack awkwardly gesticulates his surprise to see Fletcher walking down the street. He runs outside to greet him warmly.

Is it worth it for him to spend half an hour to allow the conversation to develop organically?
- No
- Yes

(a) Reciprocity First Action



Mack calls the staff together and instructs them to clean up and then go home. He grumbles to himself about the damage Fletcher has done to his business. There's no point in him keeping his staff around if there aren't any customers. Mack busied himself with some cleaning in the kitchen and managerial tasks. Before he knew it the staff had all gone home for the day.

The streets outside were still oddly empty. Mack sits at a table in the front sipping a glass of wine. A small figure appears off in the distance, but Mack doesn't give it a second thought. Anger sears his veins. Mack was accustomed to shrugging off Fletcher's insults, but things had gone too far. To attack a chef's credibility is to attack his livelihood. The figure on the street grows as it approaches. But what can Mack do? What can a humble chef do when a critic rakes him over the coals? The figure has a face. Mack awkwardly gesticulates his surprise to see Fletcher walking down the street. He runs outside to greet him warmly.

Should Mack lure Fletcher back to the restaurant or let the conversation develop on it's own?
- Mack should lure Fletcher into his restaurant.
- Mack should ignore his anger and allow the conversation to develop organically.

submit

(b) Reciprocity After Actions

Figure 33: A story event in the treatment condition of the second study when a reciprocity schema is used. The first reciprocity question is highlighted here for emphasis.

it was erased from the screen and the base story question was put in its place (as depicted in Figure 33(b)). If they answered “no” to the first action question, a concession was made and the player was asked “if they weren’t willing to commit half an hour, would they be willing to commit five minutes of Mack’s time” to achieve a particular goal. Again, after their answer to that question, the story question replaced it on the screen.

In speaking with our six participants, we discovered that five out of the six of them found this process of replacing questions to be confusing even though it was explained in the instructions. Further, one who did not describe the process as confusing, said they assumed the question replacement was an error as “the internet often doesn’t work properly on the first click.” Therefore, we believe that our design choice to strictly adhere to influence being an addition to a story event rather than a replacement for a part of the story was a cause for confusion.

Further, two study participants pointed out that the concession was meaningless to them. Both made the point that the system’s concession from “half an hour of Mack’s time” to “five minutes of Mack’s time” was still essentially instantaneous in game terms, and therefore didn’t feel like a concession at all. Despite this cost disconnect, all six of the participants agreed to the reciprocation conditions either after the first or second question; however, four out of the six participants reported not seeing any link between the reciprocity questions and the narrative question. Each of them considered the questions to be completely independent. Indeed, two out of the six participants agreed to the conditions during the reciprocal concessions and subsequently selected the inconsistent answer from the narrative question. One participant even described their answers to the reciprocity questions as atomic events that were driving the narrative—answers in the same class as the actual story answers. This feeling of disconnection between reciprocity questions and the story questions is consistent with the finding that the answer rates in the control and treatment conditions were nearly identical when reciprocity was used (see Table 18 above).

Thus, despite a lack of success in shaping the players’ decisions using reciprocity, we

Table 21: The relative frequency of general disagreement (either “strongly disagree” or “disagree”) and agreement (either “agree” or “strongly agree”) with the “I felt a sense of connection” Likert prompt in the second study. The “neither agree nor disagree” responses are not included in this table.

	Disagreement	Agreement
control	63.95%	26.74%
treatment	56.25%	31.25%

Table 22: The relative frequency of general disagreement (either “strongly disagree” or “disagree”) and agreement (either “agree” or “strongly agree”) with the “I felt a sense of engagement” Likert prompt in the second study. The “neither agree nor disagree” responses are not included in this table.

	Disagreement	Agreement
control	37.21%	47.67%
treatment	26.56%	60.94%

have an interesting result. We have a design principle for the implementation of influence schema that utilize questions: the questions should not be in addition to the action the player is to perform, but the answers to those questions should be the action the player is to perform. Additionally, leveraging costs must be relevant and tangible to players—scarcity likely works because an opportunity missed in the story is one the player perceives they will not be able to experience later. Thus, when implementing an influence schema like reciprocity, costs that are tangible to players may prove more effective.

7.8.3 Connection and Engagement

To look a bit deeper at the marginally significant findings for the connection and engagement Liker prompts, we compared the relative rates of “overall disagreement” to “overall agreement” between the conditions. To measure overall agreement or disagreement, we grouped the extremal responses together (*e.g.*, “strongly disagree” and “disagree” together comprise overall disagreement, and the same for agreement). The results are shown in Table 21 and Table 22 as the percentage of condition members falling into a particular response category. What is important to note in these two tables is that the overall disagreement is higher among control group participants than treatment group participants in both cases. These results indicate that those participants who experienced influence reported more agreement with statements about feelings of engagement with the system and feelings of a connection with the main character. Bearing in mind that the nature of influence is such that it taps into powerful behavioral tendencies, these findings could be explained by such tendencies. As influence unlocks the subconscious of people to make decisions consistent with their own behavior in the real world, they may increase their sense of self as a character in the story, resulting in increased feelings of engagement and connection. To truly understand and characterize the causes for these findings is well beyond the scope of this dissertation.

One peculiar finding we encountered in the analysis of our exit survey data had to do with the Locus of Control scores. In particular, we found that 86 of the participants in the control group completed the Likert prompts, but only 75 of those also completed the Locus of Control part of the exit survey. The fact that 11 participants did not answer all 13 Locus of Control questions but did answer the five Likert prompts is made even more interesting when we consider that the Likert prompts came *after* the Locus of Control questions—the 11 participants skipped at least one of the Locus of Control questions and continued to answer all of the Likert prompts below. On the other hand, of the 66 participants in the treatment group who completed the Likert prompts, none of them failed to complete the 13 Locus of Control questions. We found this difference to be significant with $p = 0.0025$. We can only hypothesize as to the cause of this difference. One possible explanation is that the higher levels of engagement reported by the members of the treatment group were indicative of their willingness to stay focused and answer all the exit survey questions. Fully understanding the potential causes of this effect is not possible given the design of our study, and could form an interesting direction for future research.

7.9 Conclusions

In this chapter, we have argued for the effectiveness of using influence as a tool to shape players' experiences in interactive settings. We have presented the results of two studies quantifying the effects of hand-authored scarcity statements and system generated scarcity and reciprocity statements. We have found that there is a measurable difference in players' decisions when scarcity influence statements are used and that the difference is statistically significant. We have also presented data that describes the effect of the influence statements on the players' perceptions of their experience. In four out of the five categories surveyed during the first study, we failed to find a statistically significant difference between the control group and the treatment group. In all five of the categories surveyed during the second study, we also failed to find a statistically significant difference in players' responses. In the one category that did show a statistical difference from the first study, we have presented two plausible explanations for the cause of that difference.

The completion of the second study and subsequent analyses represents, to our knowledge, the first successful implementation and evaluation of social psychological influence to shape player experiences in an interactive story and the first known drama management system to automatically generate and refine actions during episodes. Overall, the findings from the end-to-end evaluation were extremely positive. While not every aspect of our system performed as intended, every result was extremely informative even if not expected. Having completed this evaluation, we unsurprisingly have a number of open questions. More importantly, however, we have identified a few interesting design principles for future implementations of our techniques. Here, we will summarize our findings and briefly discuss some of the implications for future directions. The bulk of our discussion on future directions will be presented in Chapter 8.

After conducting two studies with hand-authored influence statements and an end-to-end evaluation of TTD-SMDPs, influence schemata, and natural language templates for a DODM drama manager in an interactive choose-your-own-adventure story, we have come to the following conclusions:

- TTD-SMDPs, even with a few ineffective actions, are an effective tool for shaping the distribution of player experiences in an interactive story
- Social psychological influence can be an effective tool for shaping player decisions in an interactive storytelling environment
- Influence schemata can be combined with natural language templates to generate and refine drama manager actions for an interactive story
- When implementing influence, care must be taken to ensure that appropriate triggers that are pertinent to the player and are tangible are used
- To reap the benefits of influence that is implemented with a give-and-take approach, relying on consistency to carry over after the influence can be detrimental to effectiveness
- Players do not perceive a change in their sense of control (and by extension self-agency) over the story as a result of influence being applied unless the influence effects steer them toward a part of the story that affects their sense of control
- The use of influence can increase players' senses of engagement and connection with the system

CHAPTER VIII

CONCLUSIONS AND FUTURE DIRECTIONS

In this work, we have designed solutions to a number of problems associated with interactive experiences that serve the needs of three types of constituents: authors/designers, technologists, and players. In particular, we are motivated by the observations:

- that having *technologists* implement computational models of social psychology concepts, we can shape *players'* decisions according to the aesthetic goals provided to drama managers by *authors*
- that we can automate solutions to key drama management design problems, thus, largely reducing the burden on authors to solve these problems while still taking into account the aesthetic goals the author prescribes for the experience

We note that, to our knowledge, no system before ours has realized either of these motivations. In order to realize these goals, we have ***designed***, ***implemented*** and ***evaluated*** a variety of artifacts. In particular, using methods of AI/ML, social psychology, and HCI we have designed, implemented and evaluated a DODM drama manager that uses innovative TTD-MDP and TTD-SMDP methods for reasoning about narrative structure, that uses social psychology concepts for reasoning about how to shape player experiences, and that automatically generates and refines actions to realize the desired changes in the decisions players make during their experiences.

In Table 23 we present a summary of the four main classifications of the contributions of this dissertation. The entries in the table cells describe the results produced to solve the problems in that category, the constituents considered when designing the solutions, and the techniques used to evaluate those solutions. For example, the design and implementation

Table 23: A brief summary of the components of this dissertation. Note that "CYOA" refers to our choose-your-own-adventure storytelling environment.

		Design & Implementation	Evaluation
Theory	**Contribution**	*DODM, TTD-MDP, TTD-SMDP Influence Schemata, templates*	*Simulation*
	Constituent	*Technologist, Author*	*Author, Player*
	Technique	*AI/ML, Social Psych*	*AI/ML*
Platform	**Contribution**	*CYOA system*	*Pilot study, end-to-end evaluation*
	Constituent	*Player*	*Author, Player*
	Technique	*HCI*	*HCI*

of our platform resulted in the choose-your-own-adventure storytelling system that was developed with the player in mind under principles we learned from the human-computer interaction field.

In the remainder of this chapter, we will discuss the contributions and results presented in the earlier chapters of this dissertation. We will describe each of the entries in Table 23 in more detail below, highlighting the specific results. Later in this chapter, we will discuss a number of future directions for this research.

8.1 Recap: Three Design Problems

Aside from implementing the environment and story itself, three design problems must be solved to fully implement a drama management system:

1. **Goal Selection**: The system must have a way of representing the state of the narrative and encoding (in terms of goals) the author's desired aesthetics for the experience. In addition, the system must have a way to reason about the player's behavior in the environment in order to select the appropriate narrative goals

2. **Action/Plan Selection/Generation**: The system must have actions that provide it a way to affect the environment. More importantly, the system must be able to reason about how the actions it takes will affect both the player's experience and its ability to achieve the gaols the author specified for it

3. **Action/Plan Refinement**: The system must have a way to ensure consistency of the actions it takes given the current state of the environment

To date, the bulk of work on drama management has been focused on the goal selection problem. Various approaches to solving that problem have been designed and to varying degrees implemented and tested in simulation or with actual game environments. Despite the significant representational and computational power provided by those approaches, the systems have relied heavily on the author to implement solutions to the action/plan selection/generation and refinement problems. In this dissertation, we presented the design of algorithms for a drama management system that automates the solutions to all three of these problems—to our knowledge the first system to do so. Our approach is based on ***computational models of influence*** which allow drama management systems to reason about how to shape players' experiences and automatically create utterances that are both meaningful in the environment and persuade the players to behave in a manner consistent with the goals the author has specified for the drama manager.

8.2 Design and Implementation

While there are a number of technical hurdles related to the implementation of environments for interactive experiences, they are orthogonal to the problems we describe and provide solutions to. In this section, we will summarize the parts of this dissertation that pertain to the design of artificial intelligence algorithms and theory that enable our drama manager to function. Additionally, we will summarize the design of our evaluation framework—a web-based choose-your-own-adventure-style storytelling system.

8.2.1 Technical Design

There are a number of technical components to this dissertation. First, there is the TTD-MDP formalism and algorithms we developed for solving them. TTD-MDPs were developed as a solution to a particular technical problem that arose from earlier work on the

Declarative Optimization-based Drama Management framework. Additionally, we developed theory to describe the performance characteristics and guarantees of the algorithms we developed for DODM. The following contributions of this dissertation comprise our solution to the first DM design problem:

- Three algorithms to solve for a TTD-MDP policy: 1) a fast linear algebra approximation, an optimization method to minimize L_1 error, and a provably globally optimal convex optimization to minimize KL-divergence
- Theoretical performance guarantees about the optimality of our approaches
- Two paradigms for authoring target distributions—encoding the author's goals for a narrative experience—in the TTD-MDP paradigm

Our influence models are the next piece of the experience management puzzle presented in this dissertation. Acknowledging the abstract nature of the DODM formalism and TTD-MDPs, we have designed a set of models of influence from social psychology to complement the abstract work on DODM. These models are a set of schemata that function as a "pluggable module" for the DODM formalism, comprising a set of dynamically generated actions the drama management system can implement without pre-specification by the author.

The application of these schemata provides utterance structures for use in the story. To realize these utterances in the narrative environment, we implemented template models that provide a variety of options for the system to generate text consistent with the influence structures. The following contributions of this dissertation comprise our solution to the second and third DM design problems:

- An organization and operationalization of six influence principles from social psychology [26] for use in interactive narratives
- A formalism based on ADL schemata [41] for concisely describing influence approaches in a form amenable to computational leverage

- A set of natural language templates for refining influence schemata into a form usable in a text-based interactive story

- A representation of drama manager actions that enable off-the-shelf algorithms to be used to solve the action/plan selection/generation and action/plan refinement problems

Because the influence models are not "atomic" actions, an adaption of the DODM/TTD-MDP formalism needed to be developed. Thus, we presented theory pertaining to modifications made to the TTD-MDP framework to allow for the use of our models. That theory, presented in Chapter 5, describes how the schema and template models can be described in terms of plans, and further, how considering drama manager actions as (possibly) non-atomic plans can be modeled effectively with an adaptation of the reinforcement learning "options" framework [123]. This insight gives rise to the TTD-SMDP formalism—another contribution of this dissertation.

8.2.2 Framework Implementation

Although not technical in the sense of an AI algorithm, the web-based storytelling system described in Chapter 7 did require some engineering. Some lightweight web-programming using css stylesheets and Java Server Pages was sufficient to create the frontend. Both for assembling story data to present it to the player as well as for storing player survey responses and story traces, the system used the Java database connection framework to connect to a MySQL database backend.

The end result of this engineering effort was a lightweight web-based storytelling system. We found the framework to be more than sufficient for gathering meaningful research data. Furthermore, the use of this framework to evaluate our complete approach—a DODM drama manager with a TTD-SMDP model, influence schemata, and natural language templates to solve each of the three DM design problems—is, to our knowledge, the first system to implement and test automated solutions to all of the drama management design problems. As the framework was by necessity engineered with certain simplifying assumptions, later in this chapter we will discuss as future work in this research program, some of the technical challenges to relaxing the appropriate assumptions.

8.3 Evaluation

While the development of theories, models, and algorithms for interactive experiences is important, the development alone does not indicate that there is utility to our methods. In order to ensure that we achieved our goals and, perhaps more importantly, that we are providing some value to authors, we conducted an evaluation of the various aspects of our solutions. There was no one-size-fits-all evaluation that we could perform. Instead, we devised different types of evaluations to highlight the different characteristics of the various contributions of this dissertation.

Loosely speaking, we have divided our evaluation into two categories: simulation and user-studies. In the past, simulation has proven to be a useful tool for understanding and characterizing the performance of various DODM solution techniques including TTD-MDPs (*cf.* [15, 21, 93, 112]); however, as pointed out by Rowe *et al.* [117], these so-called "director-centric" studies only reveal one piece of the puzzle. In addition, we must consider "cognitive-affective studies" which characterize our technologies using experiments with actual players. Another contribution of this dissertation is a set of director-centric and a set of cognitive-affective studies.

8.3.1 Technical Evaluation

We have conducted a number of evaluations of our work on the DODM formalism that have fallen into two main categories. The first category can be thought of as what Rowe would consider director-centric studies [117] and the second category is algorithmic performance. In performing both types of evaluations, we have been able to characterize the effectiveness of solutions to a TTD-MDP formulation of DODM relative to other formulations of DODM [112, 94]. Further, we have been able to make concrete claims about the specific performance of TTD-MDP solution techniques and verify those both theoretically and empirically.

8.3.1.1 *Director-centric Evaluation*

Following methods originally used by Weyhrauch as a tool to characterize the performance of his SAS+ search algorithm for DODM [135], we have conducted extensive performance studies of our TTD-MDP algorithms by comparing histograms of story qualities. We have published such comparisons extensively [15, 21, 22, 95, 96, 108, 109, 112] and presented a sample of those results in Chapter 6.

To perform these comparisons, we rely on the fact that the author's evaluation function is integral to the DODM formalism. Given a story (either sampled or resulting from a player's interaction with a storytelling system), the author's evaluation function provides us with a measure of its quality. To construct a histogram of story qualities, we need only obtain a sufficiently large set of stories. By counting the frequency with which each evaluation level occurs in the set of stories, we can create the histogram.

Using multiple sets of stories obtained under varying circumstances, we can begin to characterize the effects of different techniques. The findings we uncovered in our simulation results are qualitative in nature (not yet based on quantitative comparisons) and can be summarized as follows:

- In sufficiently large domains (*i.e.*, a reasonably sized interactive story) a sampled trajectory tree alone does not provide enough coverage for a TTD-MDP to perform effectively and, therefore, an online recovery technique may be necessary

- When authoring using a sampling approach to generate prototypes for the prototype-distance target distribution, a uniform sampling approach performs better than a rejection sampling approach, and does so with fewer samples

- TTD-MDPs with a multi-variate Gauassian mixture target distribution can be effective at balancing between different dimensions of distance

- In cases where players frequently, but not always, listen to TTD-MDP actions their overall experience can be increased by a significant reduction of "bad things" in a story accompanying only a minor reduction in "good things"

8.3.1.2 Algorithmic Performance

Aside from the director-centric studies we performed to evaluate the performance of TTD-MDPs for solving a DODM instance, we also examined the computational characteristics of the solution methods for TTD-MDPs. In particular, we compared "error" measures of three different TTD-MDP solution methods. We also compared the performance of the solution techniques in three different domains.

To perform these comparisons, we looked at the each of the "local computations" performed in sequence to solve a TTD-MDP during an episode (Section 3.3.3). Of the three techniques we developed for solving TTD-MDPs, the linear algebra approximation performs well in practice and is highly efficient computationally whereas the others provide theoretical optimality guarantees but can be slower in practice. The KL-optimal approach we have presented in this dissertation has the strongest theoretical guarantees and has shown to be more than quick enough to meet the real-time needs of an interactive experience (0.96ms for the KL-optimal approach *versus* 0.04ms for the linear algebra approximation approach).

Regarding performance of the algorithms in terms of error, we found that in almost all situations all three variants of the algorithm achieve the desired result. In certain *very* rare boundary cases, we found that the KL-optimal approach was able to find the optimal solution when the linear algebra approximation was unable to. Thus, to sum, the analyses of algorithm performance tests indicated:

- While there are no theoretical guarantees that the linear algebra approximation will result in an optimal policy, in practice the solution it finds is almost always optimal

- Although there is a penalty in computation time to perform the provably globally optimal KL minimization procedure, in practice this procedure is still well within the time allotted for a real-time interactive system

8.3.2 Human Evaluations

Fundamentally, the interactive quality that defines the artifacts we are interested in makes their evaluation impossible without human participants. Concepts such as self-agency and event realization can not be fully evaluated in simulation alone. The results of the user studies we conducted can be characterized in two groups: cognitive-affective studies and director-centric evaluations with human subjects. Cognitive-affective studies are human-focused evaluations of narrative systems. Cognitive-affective studies are those studies designed to determine player reactions to interactive experiences. In this section, we will discuss the cognitive-affective studies and human-subject evaluations we performed using our choose-your-own-adventure storytelling system.

8.3.2.1 Cognitive-affective Studies

The type of experience we seek to create for players should endow them with a sense of control. The player will, through the expression of their self-agency, be able to interact in a meaningful way with the narrative environment. In doing so, they will exert their own influence over how the progression of events occurs. In order to evaluate if we are successful in creating such an experience, we conducted human-subjects experiments and analyzed survey responses.

The player evaluation experiments we conducted utilize the choose-your-own-adventure storytelling system we have implemented. As discussed in Chapter 7, we designed this system specifically to mitigate (to the degree possible) the number of confounding factors that can affect our results and used it to perform two studies: a study focused on the use of hand-authored influence and an end-to-end evaluation of our complete approach. The findings of those cognitive-affective studies are as follows:

- Players do not perceive a change in their sense of control (and by extension self-agency) over the story as a result of influence being applied unless the influence effects steer them toward a part of the story that affects their sense of control

- The use of influence can increase players' senses of engagement and connection with the system

8.3.2.2 Human-subjects Evaluations

As discussed in Section 7.1.3 of Chapter 7, there are two types of data we collected during our studies: surveys and traces. The surveys were used to draw conclusions of the cognitive-affective variety. On the other hand, the results of the user studies we conducted that were measured directly, rather than through survey instruments, can be thought of as director-centric evaluations with human subjects. These results were obtained by examining trace information about the stories players co-created with the drama manager.

Using trace information about stories, we were able to characterize a number of interesting aspects of our approach as a whole, including if the simulation results could be verified in a fully interactive domain. As such, we analyzed the distributions of complete stories relative to each other and to the target distribution. Additionally, using this trace information, we were able to characterize the performance of influence schemata as DM actions. Briefly, the findings of our director-centric human-subjects evaluations are as follows:

- TTD-SMDPs, even with a few ineffective actions, are an effective tool for shaping the distribution of player experiences in an interactive story

- Social psychological influence can be an effective tool for shaping player decisions in an interactive storytelling environment

- Natural language templates can be combined with influence schemata to produce influence statements for an interactive story

- When implementing influence, care must be taken to ensure that appropriate triggers that are pertinent to the player and are tangible are used
- To reap the benefits of influence that is implemented with a give-and-take approach, relying on consistency to carry over after the influence can be detrimental to effectiveness

8.4 Future Directions

Following the algorithmic and evaluative contributions of this dissertation, there are numerous open questions that form possible avenues of future directions for this research program. In this section we will briefly describe some of the directions. There are far too many to list them all, so here we will focus on four of the biggest (and arguably most important): *continued algorithmic development* to ease the technical expertise needed for authoring, a more thorough *understanding and evaluation of influence for storytelling*, a thorough *evaluation of authors' abilities to leverage our technical tools*, and *relaxations to complexity reducing assumptions* made in the design of our evaluation environment.

8.4.1 Ease of Authoring

An evaluation of authors' abilities to generate stories using our paradigm is a highly desirable future direction for this work. While there have been a large number of systems for drama management of interactive stories developed, the only published literature describing authors' experiences writing interactive stories come from computer science researchers describing their efforts in implementing research systems [79, 121, 122] and not from professional or hobbyist authors.

Much of our work has been motived by the desire to help authors: 1) create increasingly complex experiences; while 2) maintaining or reducing the effort required to implement those experiences. While we have not evaluated the efficacy of our techniques at achieving those two goals thus far, intuitively we feel that we have made strides in that direction through the creation of the prototype-distance authoring paradigm for TTD-MDPs and through the development of our computational models of influence. Approaches to verifying that we have indeed simplified the authoring process will require potentially extensive discussions and observations of authors creating interactive experiences in our paradigm. The goals of these studies would not be an examination of the end result of the authoring process, but a study of the process itself. Three examples of the types of questions we would seek to answer are: "Can authors describe stories in terms of plot graphs with precedence constraints?"; "Are authors comfortable specifying distance measures for target distributions?"; and "Can authors comfortably specify appropriate natural language templates for refining influence schemata effectively?" The results of pursuing this direction of research would help to focus technical work on algorithmic changes that may still be necessary in order for these approaches to be useful for professional and hobbyist authors alike.

8.4.2 Interface Evaluation for Authors

Because directly updating database tables to create story events and transitions based on the answers to specific questions is quite burdensome, we implemented an authoring tool to ease our authoring process. The tool provides a graphical representation of story events, transitions between them, and a way to update the associated text, video, question, and answer data. The story is encoded by the author as a directed acyclic graph where vertices represent story events and have associated pre-text, video, post-text, and question entries. Edges represent transitions and map a story event and a question answer to the next story event.[1] The graph depiction of part of the story used in our study is presented in Figure 34.

To edit the story information associated with an event or a transition, there is an editor pane that allows editing entries and event associations with those entries. By changing a

[1]The concept of a GUI story editor is not new (*cf.* [81]).

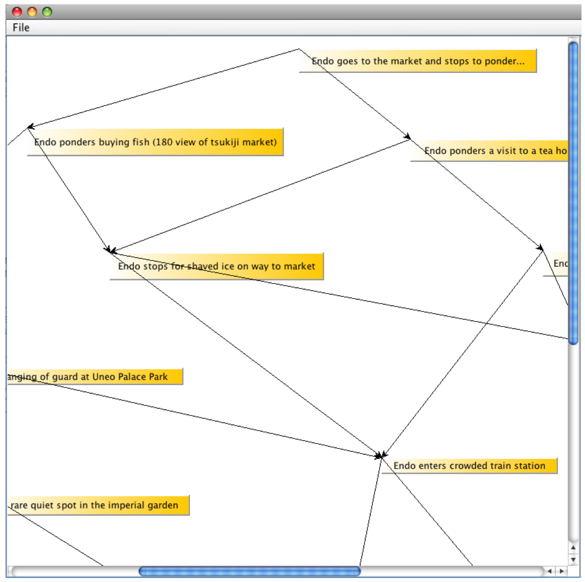


Figure 34: The graphical representation of the story structure in the authoring tool. The vertices represent story events and the edges represent the possible transitions between the story events.

value in a form, all of the appropriate updates are made to the relevant database tables. Also, this approach allows for recycling text or videos across multiple events. If the author wishes to use a question in the event they are currently editing the question has been entered into the database earlier, they can simply enter the unique identifier assigned by the database and all the relevant fields for the event will be updated. The design of the authoring tool was sufficient for the purposes of creating a story for our study; however, it has not been evaluated with authors (professional or otherwise) yet. This type of evaluation would be distinct from those design evaluations discussed in Section 8.4.1. In the design evaluations we would seek to understand the authoring *process* better, trying to modify our technical solutions to best fit the needs of authors. On the other hand, these interface evaluations would focus on the *end result*, trying to design software tools that work in harmony with our technical solutions to enable authors to fully leverage the power of our algorithms and their own creative talent.

8.4.3 Influence Studies

The results of the user studies described in this thesis shed some light on the function of social psychological influence in a storytelling setting. We confirmed that scarcity can be an effective tool at shaping player decisions in forced-choice alternative settings; however, we don't yet have a solid understanding of the contexts in which it is or is not effective. We also learned about some potential pitfalls pertaining to the implementation of reciprocity. Much more needs to be learned as to whether there are contexts in which reciprocity does have a significant effect. At this point, little is known about the other four principles we surveyed in this dissertation. Additionally, little is known about the specifics of generalizing from text-based story environments to graphical or sandbox style environment where forced-choice alternative situations are less likely to be used.

Thus, another important direction for future research in this arena is to run extensive studies on how influence functions in various domains with different characteristics, storytelling or otherwise. The ultimate goal of pursuing this avenue of research would be to derive first-principles that govern the design and implementation of influence in storytelling settings. Realizing this goal will require a thorough evaluation of many different theories of influence in many different settings to learn what works, learn where it works, and try to understand why it works. The more we are able to discern from these studies, the more accurately we can develop models of influence and techniques for refining those models in various storytelling domains.

8.4.4 Relaxing Complexity-reducing Assumptions

In their AI text, Russell and Norvig [118] describe seven dimensions upon which a task domain can be characterized. Each of those dimensions can take on two possible values. The complexity of every task domain can then be described by those seven attributes. The dimensions that Russell and Norvig describe are:

- **Fully observable** vs. **partially observable**: If the environment provides access to the complete state that is relevant to decision making, the environment is said to be "fully observable", otherwise it is "partially observable"

- **Single agent** vs. **multiagent**: In the simplest case, an environment is a "multiagent" environment if there are other entities acting in the world and a "single agent" environment if there are not; however, in cases where other entities are not modeled as (or not actually) attempting to maximize a performance measure subject to another agent's behavior, we need not consider those entities as agents

- **Deterministic** vs. **stochastic**: When an action taken by an agent and the current state of the environment are the complete determiners of the next state of the environment it is said to be "deterministic." If there are external factors that can alter the outcome of actions the environment is said to be "stochastic"

- **Episodic** vs. **sequential**: In an "episodic" environment, the agent's percept-action cycle is atomic and any future cycles are completely free from influence of prior episodes. If an agent's future decisions are affected at all by its current decision, the environment is said to be "sequential"

- **Static** vs. **dynamic**: If the agent's period of deliberation occurs without change to the environment, the environment is said to be "static." If the environment can evolve while the agent is deliberating, the environment is "dynamic"

- **Discrete** vs. **continuous**: This distinction is applied to the state of the environment, the way in which time is managed, the actions available to the agent, as well as its percepts. If the state, time indices, actions, and percepts are countable (*i.e.*, can be enumerated) the environment is "discrete." If any of these things take on continuous values in a range, the environment is "continuous"

- **Known** vs. **unknown**: Pertaining to the agent's knowledge of the environment rather than to the environment itself, "known" indicates that the (probabilistic) outcomes of all actions are known. If they are not known, then the environment is "unknown" and must be learned (this is sometimes referred to as model-free)

In general, the first of the two alternatives for each dimension represents the "easier" of the two from an AI or ML design standpoint. In many cases, however, the easier values are a more accurate model of the real world the environment describes. In other cases, designers may make simplifying assumptions in order to create models of the world that are easier for agents to perceive and act in.

In the case of our work on DODM and the evaluation environments we use, we have a mixture of "hard" and "easy" attributes. In particular, the environments we are concerned with are fully observable, single agent, stochastic, sequential, static, discrete, known environments. Below is a more detailed explanation of the characteristics of the DODM formalism that lend themselves to these particular environment attributes:

- DODM is **fully observable**. The story state is fully specified by the sequence of story events and actions taken. This information is made directly available to the DODM manager

- DODM is technically a multiagent environment with the human player and drama manager interacting; however, because the human player can be modeled as stochastic (rather than maximizing performance according to some function), we consider DODM to be a **single agent** environment[2]

- DODM is **stochastic**. The player's behavior, since not maximizing performance, is modeled well (albeit not always accurately if authors' assumptions are wrong) by nondeterminism

- DODM is **sequential**. Narrative, by it's very nature, is not episodic; however, in

[2]We have performed some experiments where we have tried to model the player as maximizing an unknown evaluation function defined over features of the story the drama manager uses for evaluation [114, 115]. The results were mixed at best. Therefore, we have opted to continue to characterize players as stochastic rather than goal-seeking.

our case computation can be performed episodically as a result of theoretical guarantees provided by the online KL-divergence solution to TTD-MDPs. Note here that episodic is different than the notion of an episode discussed in relation to our user studies

- DODM can be either **static** or dynamic. For the purposes of this dissertation, we have chosen to apply DODM in a static environment
- The implementation of DODM presented in this dissertation is **discrete**. It would be possible to have continuous actions or continuous time
- DODM is implemented with a fully-specified model of player behavior, states, and actions which makes it a **known** environment

We would be remiss if we did not point out that our techniques are not a silver bullet. There are circumstances under which they will not apply. For example, the variety of influence that we have chosen to model is marked by a strong social context. We have argued that many computer games, even those not traditionally thought of as story-based like chess or poker, can be managed using the tools we have developed for drama management [113]; however, while the TTD-MDP formalism and DODM may apply to chess, it is almost certainly the case that the influence models we have developed will fail due to the lack of social context or language interaction modes. There are other theories of influence that may apply in those situations, but we have not explored them to date.

There are two main complexity reducing assumptions we could eliminate in our evaluation of DODM: we could use a *dynamic* rather than static environment; and/or we could model *continuous* state, actions, or time instead of discrete versions of all of these.

In order for our approaches to function in a dynamic environment, there are certain concepts that would need to be modeled effectively such as action failure or preemption. The current assumption of the static environment allows for the drama manager to ensure that the action it has deliberated about and chosen to perform will be applicable in the environment. In developing TTD-SMDPs (Chapter 5), we included a model of action failure, but illustrated how it wasn't necessary for our simplified evaluation environment. The implementation of TTD-SMDPs would need to rely on that model in full if it were a dynamic environment.

On the other hand, in order for our approaches to function in a continuous environment, there are a few more complicated challenges that will need to be tackled. For example, in a continuous state environment, we lose the closed-loop cycle of story event → DM action → player transition → story event. To handle this situation, a model of timing constraints and story event triggers would need to be developed and integrated into the TTD-MDP formalism. Magerko's IDA (see Section 2.5.1) successfully implemented such a model [68]. Further, dealing with continuous state invites the need for a model of *proximity* to inform transitions between events as well as how actions can get refined. For example, in a continuous state sandbox style environment, the use of natural language generation to solve the action/plan refinement problem may not be appropriate. In those cases, models of physical proximity (perhaps based on principles like juxtaposition [124]) might prove useful.

8.5 Conclusions

To restate our thesis: Using concepts from narratology, interactive storytelling, and social psychology, we can design efficient algorithms and compact representations that enable computer systems to reason about narrative and shape human player experiences according to the aesthetic goals specified by authors. We have presented a number of models, algorithms, and results from our work on drama management and computational influence as well as simulation and user-study evaluations to support our thesis. To sum, the contributions of this dissertation are as follows:

Design: We have designed a number of algorithms and an evaluation framework. We developed:

- the Targeted Trajectory Distribution Markov Decision Process formalism
- three solution variants for the TTD-MDP formalism
- two authoring paradigms for the TTD-MDP formalism
- computational models of Cialdini's *click whirr* responses [26] in the form of influence schemata and template models
- a generalization of the TTD-MDP framework, called TTD-SMDPs, based on an adaptation of the MDP options framework to handle non-atomic actions
- a theory and procedure for transferring probabilistic player models across domains
- a process for automatically generating and refining DODM action using social psychology influence schemata and natural language temples—to our knowledge the first working solutions to the second and third DM design problems
- a web-based choose-your-own-adventure-style storytelling system

Evaluation: We thoroughly evaluated our algorithms and techniques using a variety of tools. We have:

- evaluated TTD-MDPs on three different "toy" domains, one representing an actual interactive story world
- characterized the performance of TTD-MDPs relative to competing solution techniques for Declarative Optimization-based Drama Management
- characterized the computational performance of the three solution techniques for TTD-MDPs, showing that the provably optimal algorithm performs best and showing that it does so in a reasonable amount of time [15]
- implemented and tested a complete DODM drama manager using TTD-MDPs
- conducted what we believe are the first ever user studies that illustrate the effectiveness of using influence models to shape player decisions in an interactive

storytelling environment, demonstrating in two separate tests that at least one method of influence has a significant effect, as well as indicating situations in which influence may fail

- derived design principles from the successes and failures of our implemented influence schemata—principles that will inform the future implementation of influence schemata in other domains
- evaluated a complete end-to-end system that implemented technical solutions to all three drama management design problems

We have developed and evaluated compact representations and efficient algorithms that enable AI systems to reason about and take action to shape player experiences in interactive narratives. Our work has shown that modeling influence from social psychology can be a powerful tool in aiding authors to construct interactive virtual experiences that conform to the aesthetic goals they specify.

APPENDIX A

QUALITATIVE ANALYSIS OF RELATED WORK SUMMARY

A.1 Desiderata Summary Table

Here, we present a table summarizing the qualitative analysis provided in the text above. It is intended for use a reference to guide the reader interested only in a few of the systems surveyed that exhibit the properties they are interested in. For ease, the order of presentation of the systems is the same order as in the body of the text.

	speed	**coord**	**replay**	**control**	**self-agency**	**authoring**	**adapt**	**sound**	**invisible**	**measure**
SBDM	○	◐	○	◐	◐	○	○	○	◐	●
DODM	●	◐	○	◐	◐	○	○	●	◐	●
TTD-MDPs	●	◐	●	◐	●	●	○	●	◐	●
Mimesis	◐	●	○	●	●	○	○	○	○	○
ASD	●	◐	○	●	●	○	○	●	◐	○
Dilemmas	○	●	●	●	●	○	●	○	●	●
IDA	○	●	○	○	●	●/ ○	○	○	●	●
U-Director	○	◐	○	◐	◐	○	●	●	◐	●
Beat-based	●	●	●	●	●	○	○	○	◐	○
OPIATE	○	●	○	●	●	○	●	○	◐	○
Preference Modeling	○	◐	○	○	◐	◐	●	○	◐	●
PaSSAGE	●	◐	○	●	●	○	●	○	●	○
Narrative Learning	○	◐	○	●	●	●	○	○	◐	○

APPENDIX B

DETAILS OF THE SURVEY DESIGN

In this appendix, we present the survey instruments used in the user studies presented in Chapter 7.

B.1 Demographic Survey

```
What is the highest level of education you have attained?
+ High school
+ Some college
+ Associate's degree
+ Bachelor's degree
+ Some graduate school
+ Master's degree (or equivalent)
+ Ph.D.
```

```
How old are you?
+ Less than 18
+ 18-24
+ 25-29
+ 30-34
+ 35-39
+ 40-44
+ 45-49
+ Greater than 50
```

```
How many hours a week do you spend playing games on the
computer (approximately)?
+ Fewer than 4
+ 5-9
+ 10-14
+ 15-19
+ 20-24
+ More than 25
```

```
How many hours a week do you spend using the internet
recreationally?
+ Fewer than 4
+ 5-9
+ 10-14
+ 15-19
+ 20-24
+ More than 25
```

B.2 Exit Survey Likert Prompts

Please indicate your level of agreement with the following five statements.

	Strongly Agree	Agree	Neither Agree Nor Disagree	Disagree	Strongly Disagree
I felt a sense of control over the story progression.	○	○	○	○	○
I felt the story was adapted to me.	○	○	○	○	○
I felt manipulated by the system.	○	○	○	○	○
I felt a connection to the character in the story.	○	○	○	○	○
I felt a sense of engagement with the system.	○	○	○	○	○

B.3 Locus of Control Questions

The 13 question locus of control scale instrument included in the exit survey of our end-to-end user study. Here, we have annotated each answer with it's score in parenthesis for reference. When given to study participants, the scores were not visible. Scoring is cumulative and a high score indicates an external locus of control whereas a low score indicates an internal locus of control.

For each of the following 13 alternatives, please choose the alternative that ***most closely*** matches your feelings about the experience you just had.

1.	○: Many of the unhappy things in Mack's life are partly due to bad luck. (1) ○: Macks's misfortunes result from the mistakes he makes. (0)
2.	○: In the long run Mack got the respect he deserved. (0) ○: Unfortunately, Mack's worth passed unrecognized no matter how hard he tried. (1)
3.	○: Without the right breaks Mack cannot be a successful chef. (1) ○: Mack is a capable chef, but has failed to take advantage of his opportunities. (0)
4.	○: No matter how hard he tries some people just don't like Mack. (1) ○: Because Mack isn't well liked, he must not understand how to get along with others. (0)
5.	○: I found that what was going to happen to Mack will happen. (1) ○: Trusting to fate wasn't going to work out for Mack, so he made a decision to take a definite course of action. (0)
6.	○: Mack becoming a success was a matter of hard work, luck had little or nothing to do with it. (0) ○: Mack's success depended mainly on being in the right place at the right time. (1)
7.	○: When Mack made plans, I was almost certain that I could make them work. (0) ○: It wasn't wise to plan too far ahead because many things turned out to be a matter of good or bad fortune. (1)
8.	○: In Mack's case getting what he wanted had little or nothing to do with luck. (0) ○: Many times he might just as well have decided what to do by flipping a coin. (1)
9.	○: As far as story events are concerned, Mack was the victim of forces I could neither understand, nor control. (1) ○: By taking an active toll, I could get Mack to control story events. (0)
10.	○: In the long run the bad things that happened to Mack are balanced by the good ones. (1) ○: Mack's misfortunes were the result of lack of ability, ignorance, laziness, or all three. (0)
11.	○: Many times I felt that I had little influence over the things that happened to Mack. (1) ○: It was impossible for me to believe that chance or luck played an important role in Mack's story. (0)
12.	○: Mack was lonely because he didn't try to be friendly. (0) ○: There wasn't much use in trying too hard to please people, if they liked Mack, they like him. (1)
13.	○: What happened to Mack was my own doing. (0) ○: Sometimes I felt that I didn't have enough control over the direction of Mack's life. (1)

APPENDIX C

DETAILED RESULTS FROM STUDY 1

Here we present the complete set of results of the demographic and exit surveys from the first study.

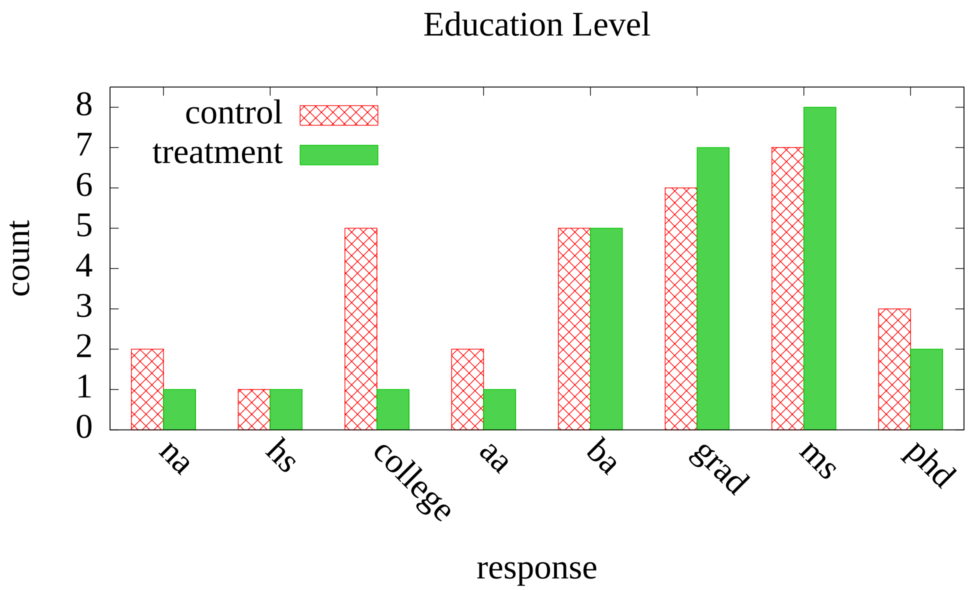


Figure 35: The distribution of responses to the demographic survey question: "What is the highest level of education you have obtained?" There were 31 participants in the control group and 26 participants in the treatment group that answered this question.

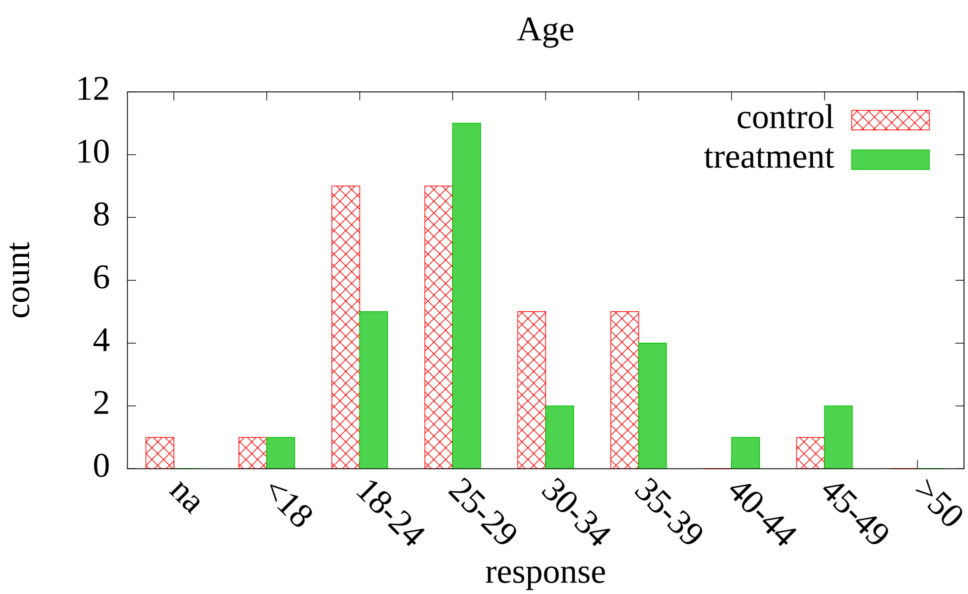


Figure 36: The distribution of responses to the demographic survey question: "How old are you?" There were 31 participants in the control group and 26 participants in the treatment group that answered this question.

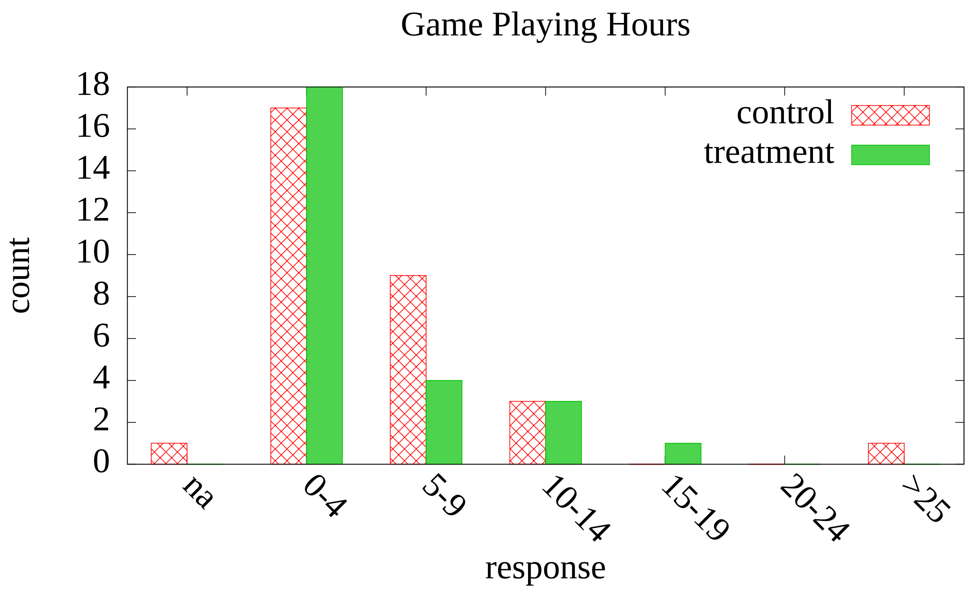


Figure 37: The distribution of responses to the demographic survey question: "How many hours a week do you spend playing games on the computer (approximately)?" There were 31 participants in the control group and 26 participants in the treatment group that answered this question.

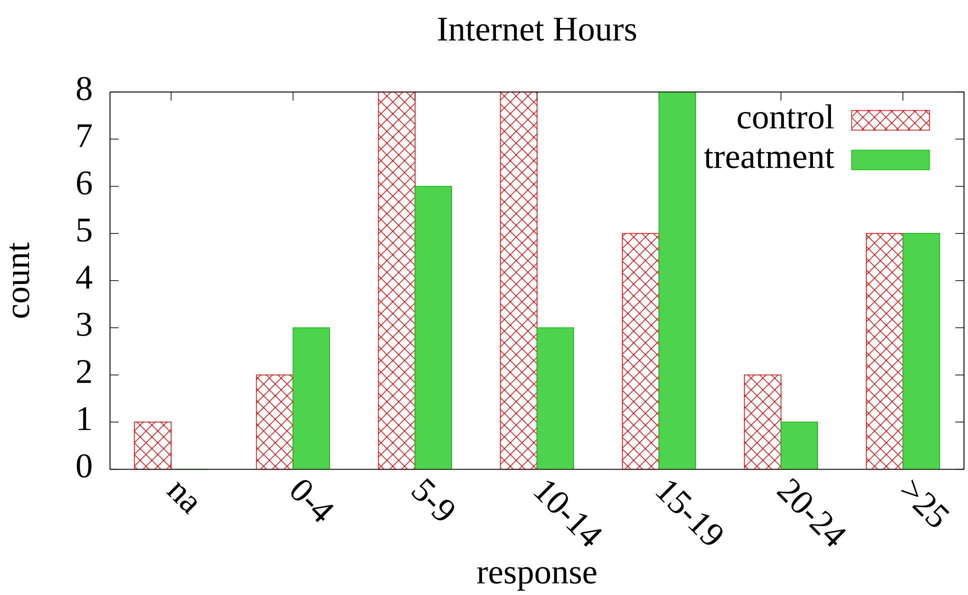


Figure 38: The distribution of responses to the demographic survey question: "How many hours a week do you spend using the internet recreationally?" There were 31 participants in the control group and 26 participants in the treatment group that answered this question.

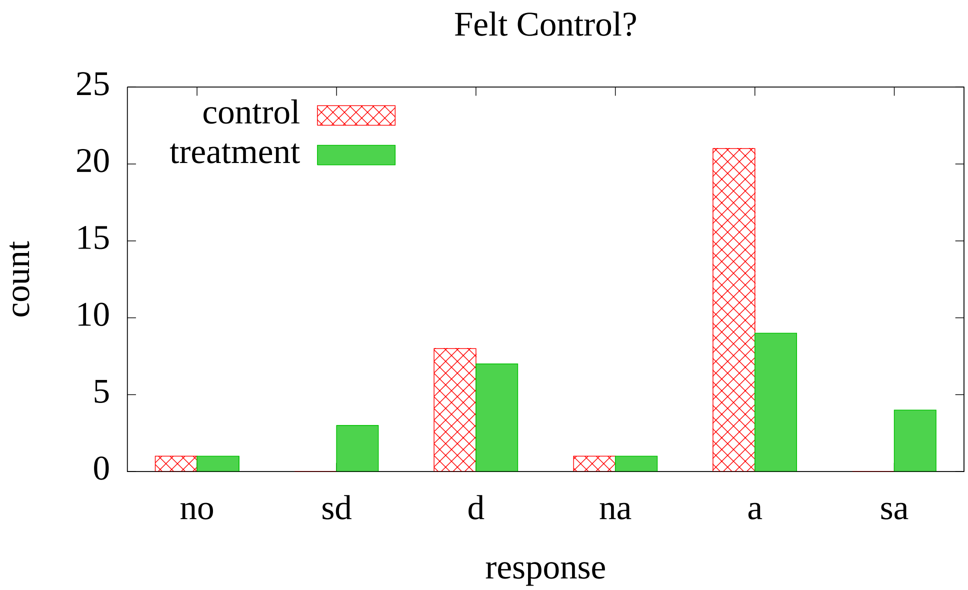


Figure 39: The distribution of responses to the Likert prompt: “I felt a sense of control over the story progression.” There were 31 participants in the control group and 25 participants in the treatment group that answered this question.

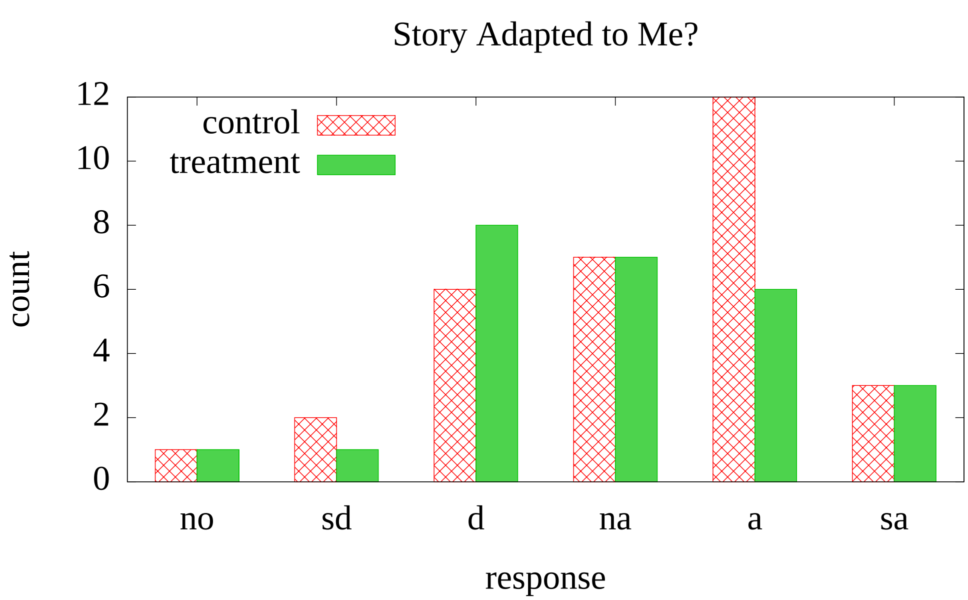


Figure 40: The distribution of responses to the Likert prompt: “I felt the story was adapted to me.” There were 31 participants in the control group and 26 participants in the treatment group that answered this question.

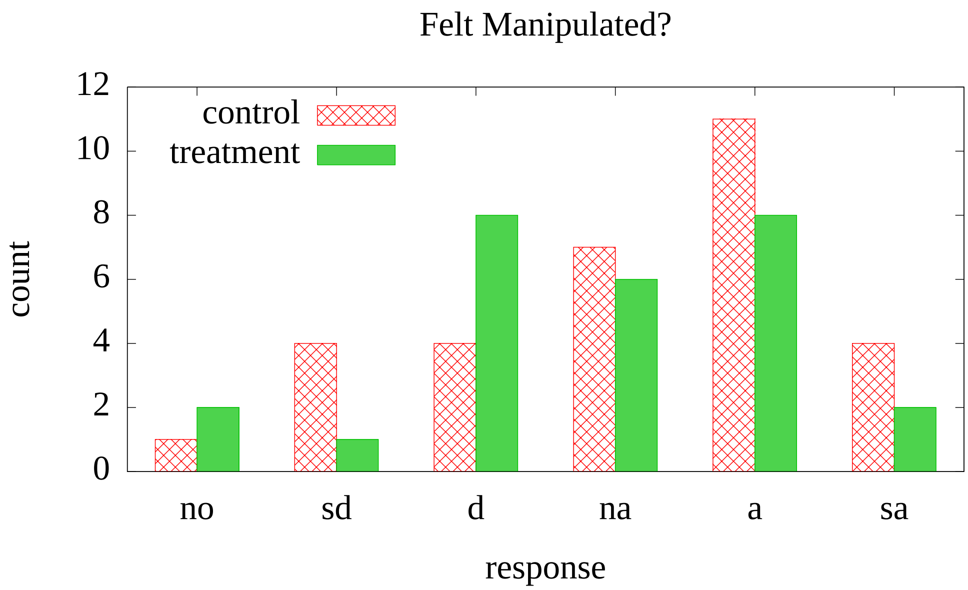


Figure 41: The distribution of responses to the Likert prompt: "I felt manipulated by the system." There were 31 participants in the control group and 27 participants in the treatment group that answered this question.

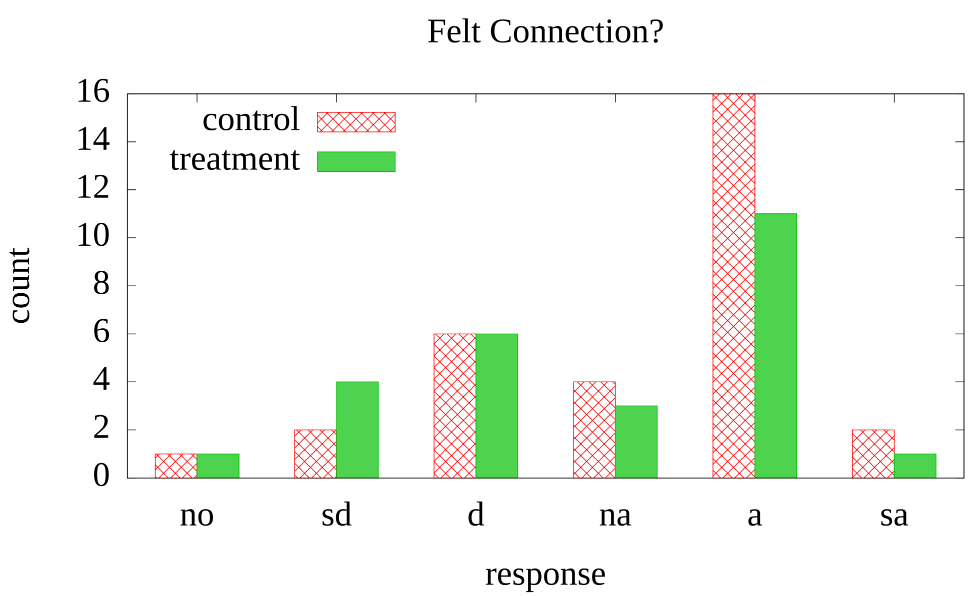


Figure 42: The distribution of responses to the Likert prompt: "I felt a sense of connection with the character in the story." There were 31 participants in the control group and 26 participants in the treatment group that answered this question.

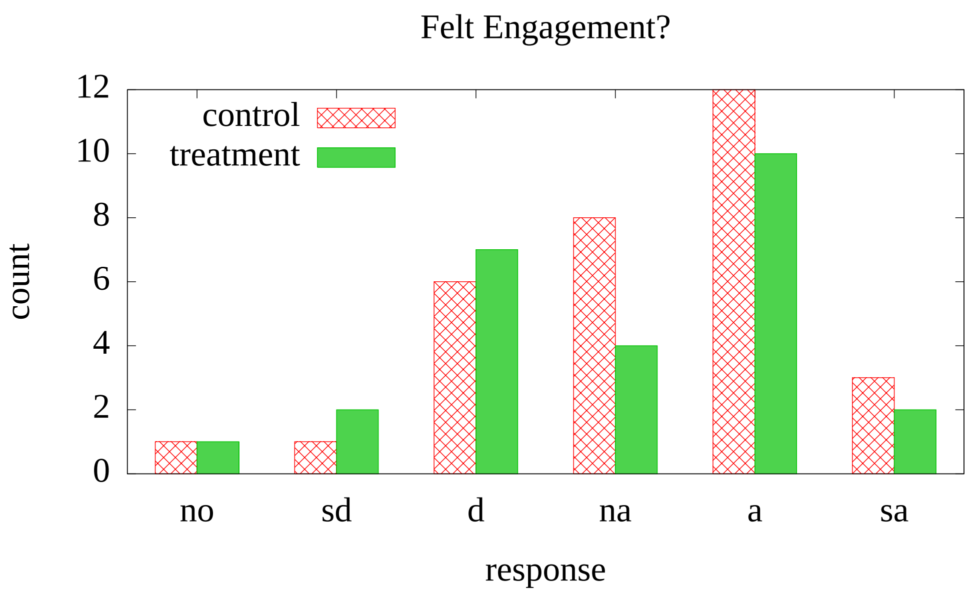


Figure 43: The distribution of responses to the Likert prompt: "I felt a sense of engagement with the system." There were 31 participants in the control group and 26 participants in the treatment group that answered this question.

APPENDIX D

DETAILED RESULTS FROM STUDY 2

Table 24: The KL-divergence statistics for the control and treatment distributions compared to the target distribution when data is accumulated across multiple episodes.

Num episodes	$D_{KL}(target\|\|control)$	$D_{KL}(target\|\|treatment)$
1	0.9657898807544473	0.510294752666971
2	1.1633593283130341	1.2561929681895827
3	1.1658786244548534	1.2721619472490655
4	1.1658786244548534	1.2812679056813108

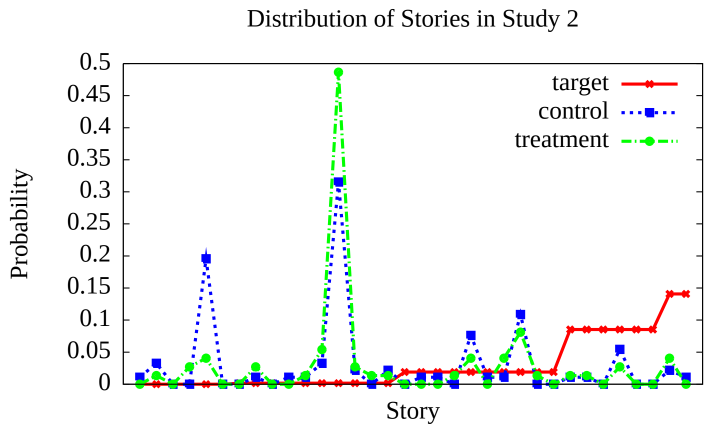


Figure 44: The full distributions of stories for the second study accumulated over *a single episode*. The x-axis is an arbitrary indexing over complete stories sorted in ascending order of target probability. The y-axis indicates the frequency that story occurred (or was targeted in the case of the target distribution). Included in this plot is the target distribution, the distribution of stories experienced by participants in the control group, and the distribution of stories experienced by participants in the treatment group.

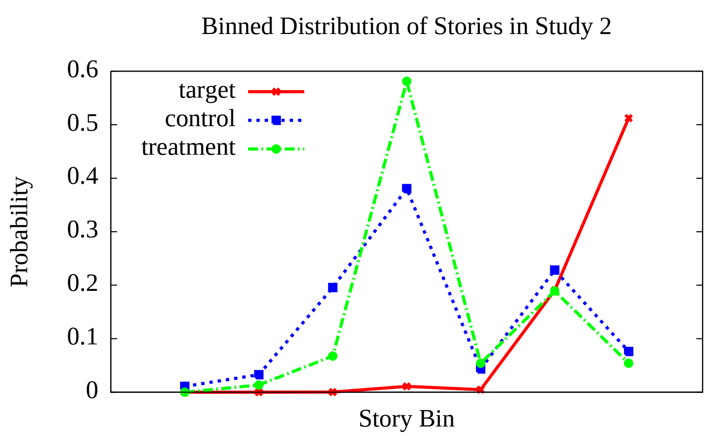


Figure 45: The full distributions of stories for the second study with "cumulative binning" accumulated over *a single episode*. The x-axis is an arbitrary indexing over complete stories sorted in ascending order of target probability. The y-axis indicates the target probability or frequency a story in that bin occurred during an episode or was targeted (in the case of the target distribution). Included in this plot is the target distribution, the distribution of stories experienced by participants in the control group, and the distribution of stories experienced by participants in the treatment group.

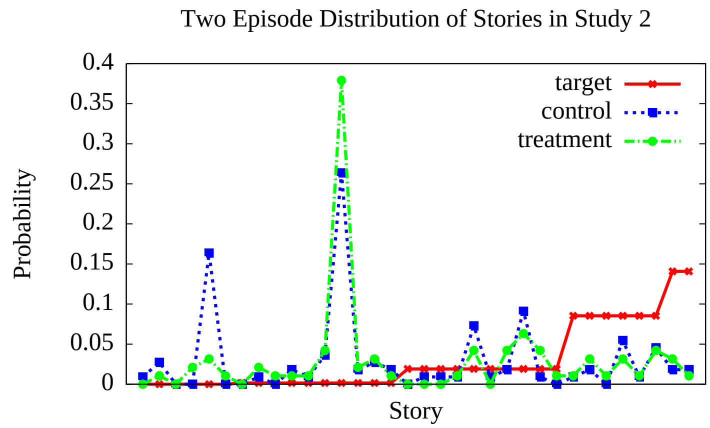


Figure 46: The full distributions of stories for the second study accumulated over *two episodes*. The x-axis is an arbitrary indexing over complete stories sorted in ascending order of target probability. The y-axis indicates the frequency that story occurred (or was targeted in the case of the target distribution). Included in this plot is the target distribution, the distribution of stories experienced by participants in the control group, and the distribution of stories experienced by participants in the treatment group.

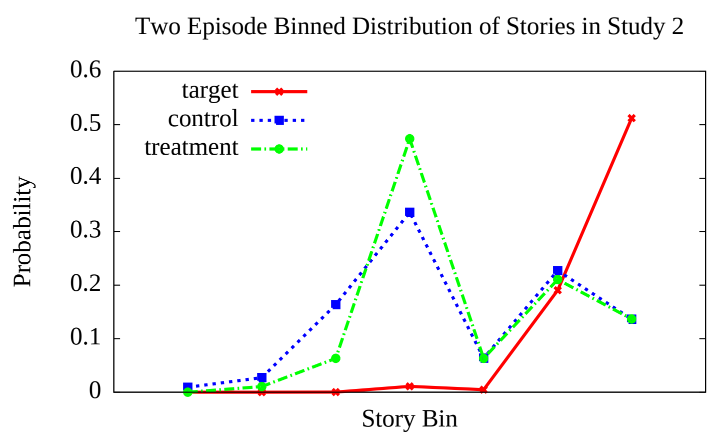


Figure 47: The full distributions of stories for the second study with "cumulative binning" accumulated over *two episodes*. The x-axis is an arbitrary indexing over complete stories sorted in ascending order of target probability. The y-axis indicates the target probability or frequency a story in that bin occurred during an episode or was targeted (in the case of the target distribution). Included in this plot is the target distribution, the distribution of stories experienced by participants in the control group, and the distribution of stories experienced by participants in the treatment group.

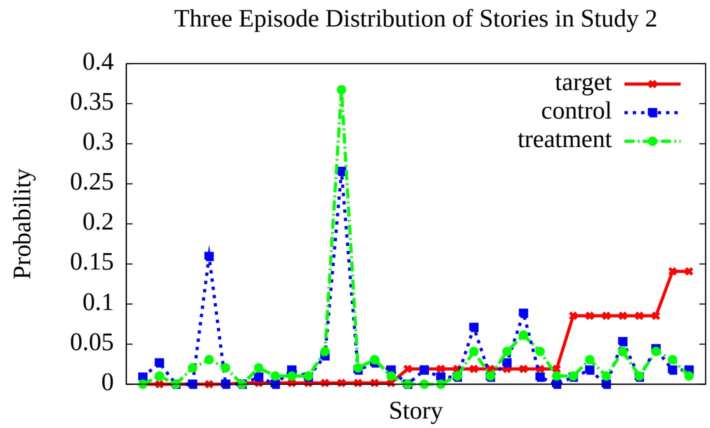


Figure 48: The full distributions of stories for the second study accumulated over *three episodes*. The x-axis is an arbitrary indexing over complete stories sorted in ascending order of target probability. The y-axis indicates the frequency that story occurred (or was targeted in the case of the target distribution). Included in this plot is the target distribution, the distribution of stories experienced by participants in the control group, and the distribution of stories experienced by participants in the treatment group.

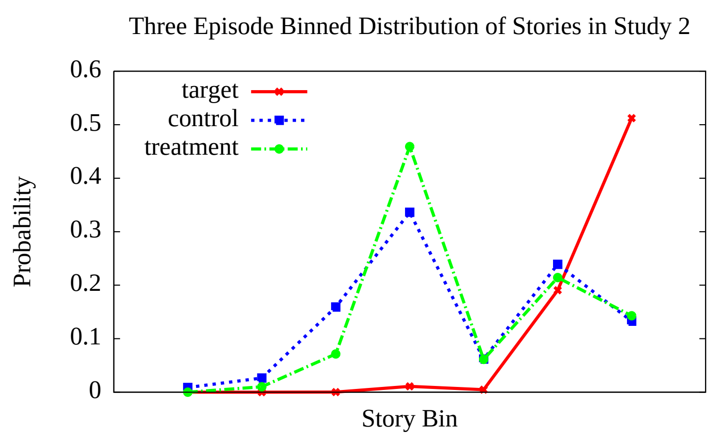


Figure 49: The full distributions of stories for the second study with "cumulative binning" accumulated over *three episodes*. The x-axis is an arbitrary indexing over complete stories sorted in ascending order of target probability. The y-axis indicates the target probability or frequency a story in that bin occurred during an episode or was targeted (in the case of the target distribution). Included in this plot is the target distribution, the distribution of stories experienced by participants in the control group, and the distribution of stories experienced by participants in the treatment group.

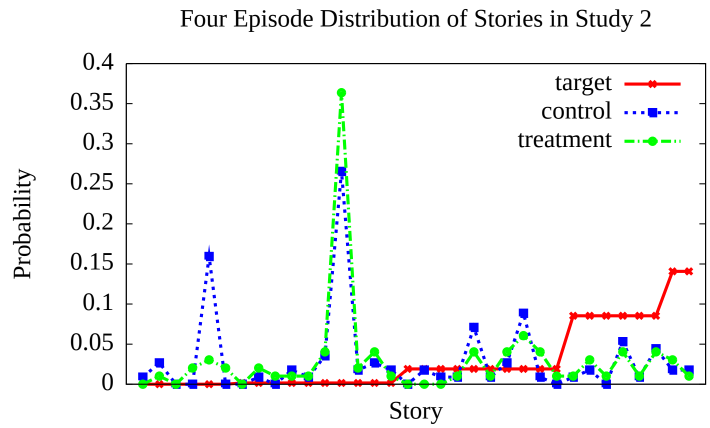


Figure 50: The full distributions of stories for the second study accumulated over *four episodes*. The x-axis is an arbitrary indexing over complete stories sorted in ascending order of target probability. The y-axis indicates the frequency that story occurred (or was targeted in the case of the target distribution). Included in this plot is the target distribution, the distribution of stories experienced by participants in the control group, and the distribution of stories experienced by participants in the treatment group.

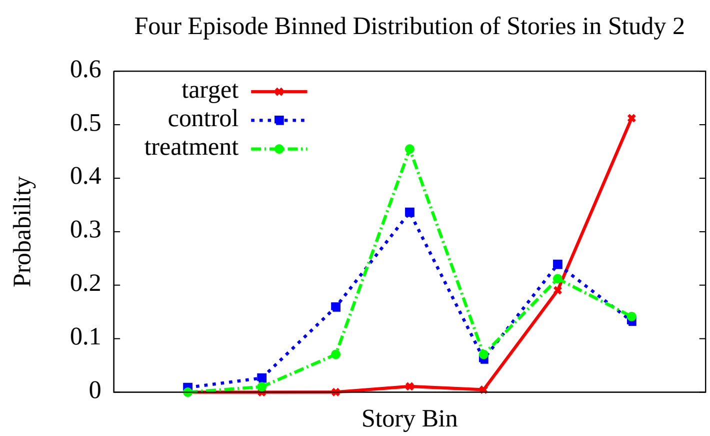


Figure 51: The full distributions of stories for the second study with "cumulative binning" accumulated over *four episodes*. The x-axis is an arbitrary indexing over complete stories sorted in ascending order of target probability. The y-axis indicates the target probability or frequency a story in that bin occurred during an episode or was targeted (in the case of the target distribution). Included in this plot is the target distribution, the distribution of stories experienced by participants in the control group, and the distribution of stories experienced by participants in the treatment group.

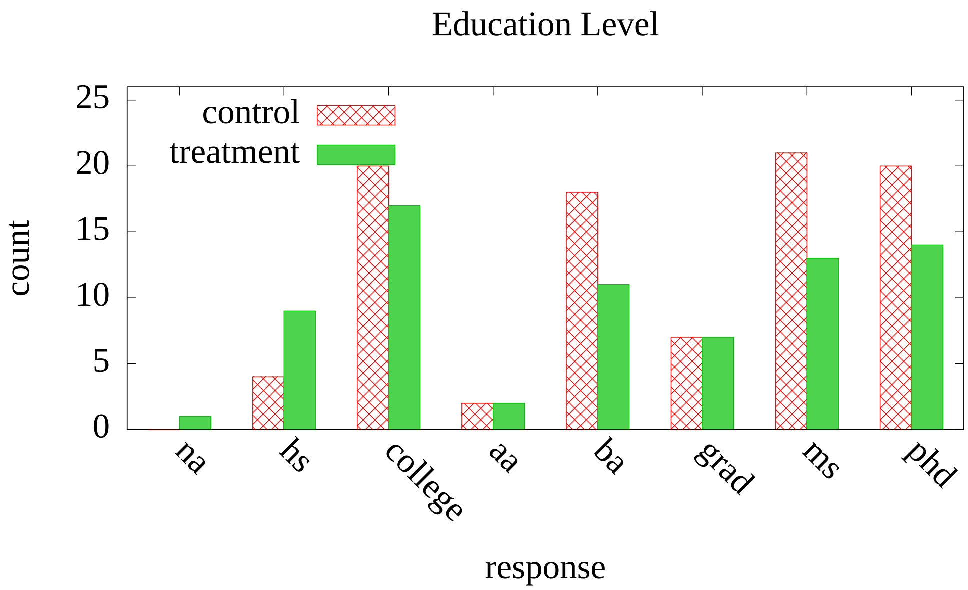


Figure 52: The distribution of responses to the demographic survey question: "What is the highest level of education you have obtained?" There were 92 participants in the control group and 74 participants in the treatment group that answered this question.

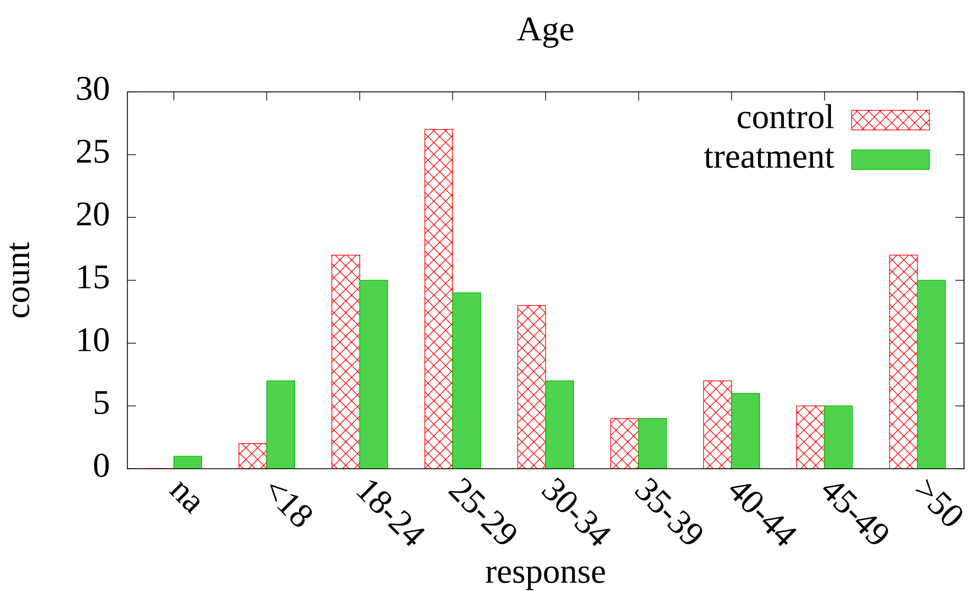


Figure 53: The distribution of responses to the demographic survey question: "How old are you?" There were 92 participants in the control group and 74 participants in the treatment group that answered this question.

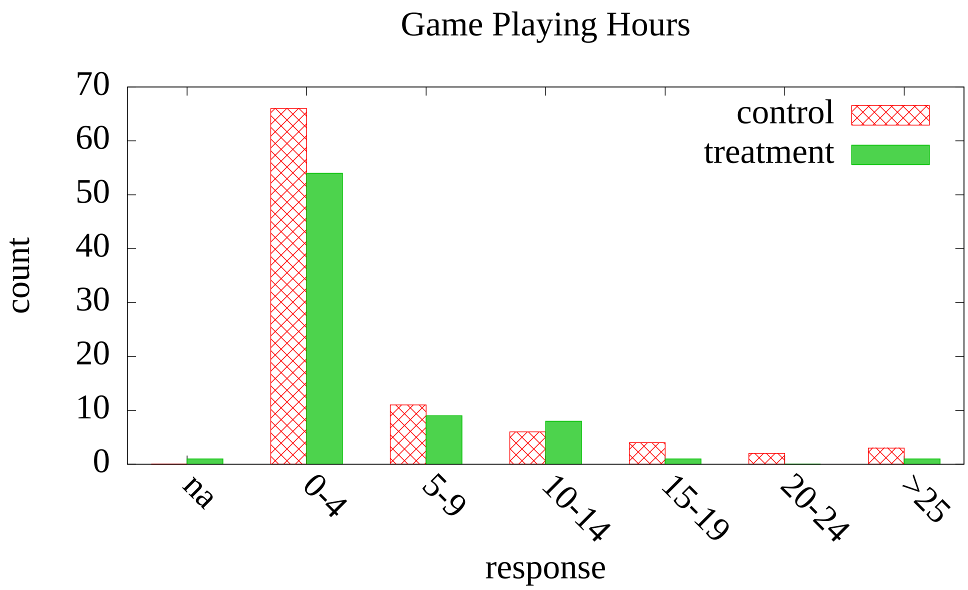


Figure 54: The distribution of responses to the demographic survey question: "How many hours a week do you spend playing games on the computer (approximately)?" There were 92 participants in the control group and 74 participants in the treatment group that answered this question.

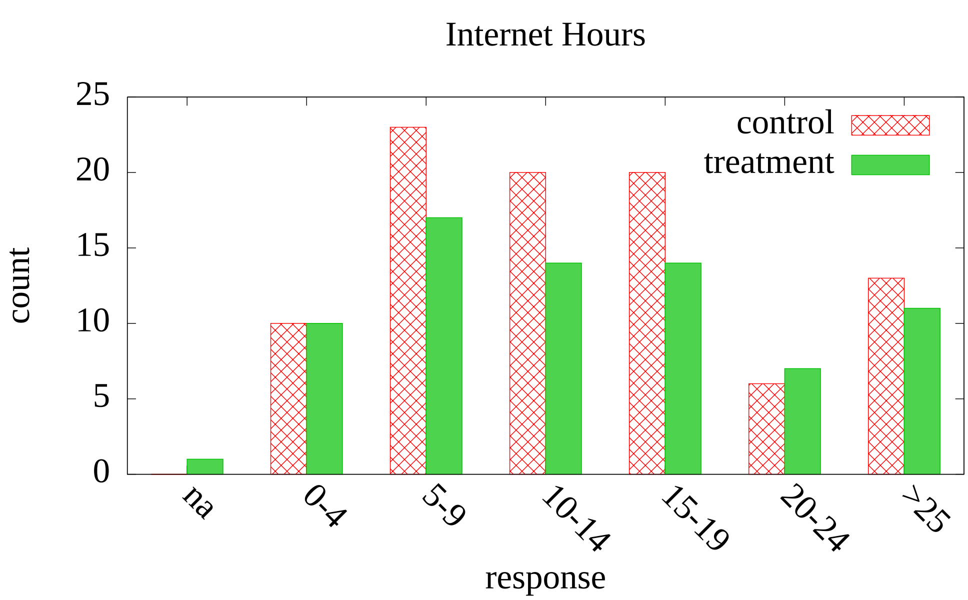


Figure 55: The distribution of responses to the demographic survey question: "How many hours a week do you spend using the internet recreationally?" There were 92 participants in the control group and 74 participants in the treatment group that answered this question.

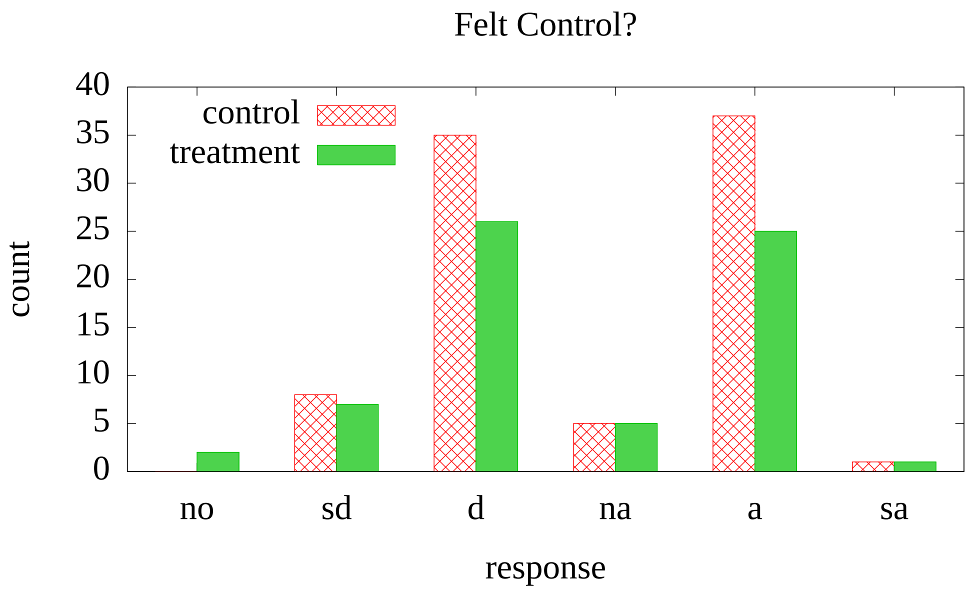


Figure 56: The distribution of responses to the Likert prompt: "I felt a sense of control over the story progression." There were 86 participants in the control group and 66 participants in the treatment group that answered this question.

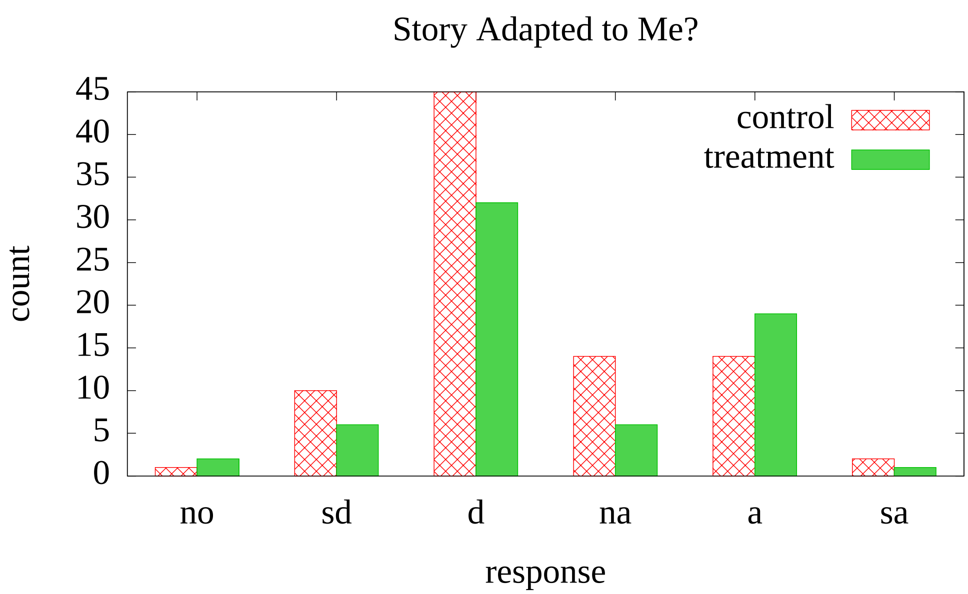


Figure 57: The distribution of responses to the Likert prompt: "I felt the story was adapted to me." There were 86 participants in the control group and 66 participants in the treatment group that answered this question.

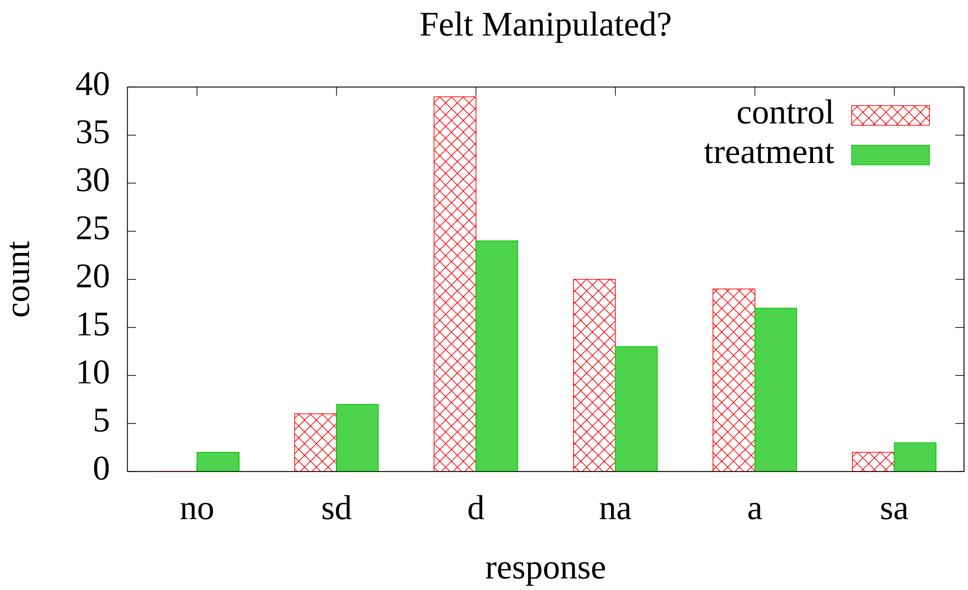


Figure 58: The distribution of responses to the Likert prompt: "I felt manipulated by the system." There were 86 participants in the control group and 66 participants in the treatment group that answered this question.

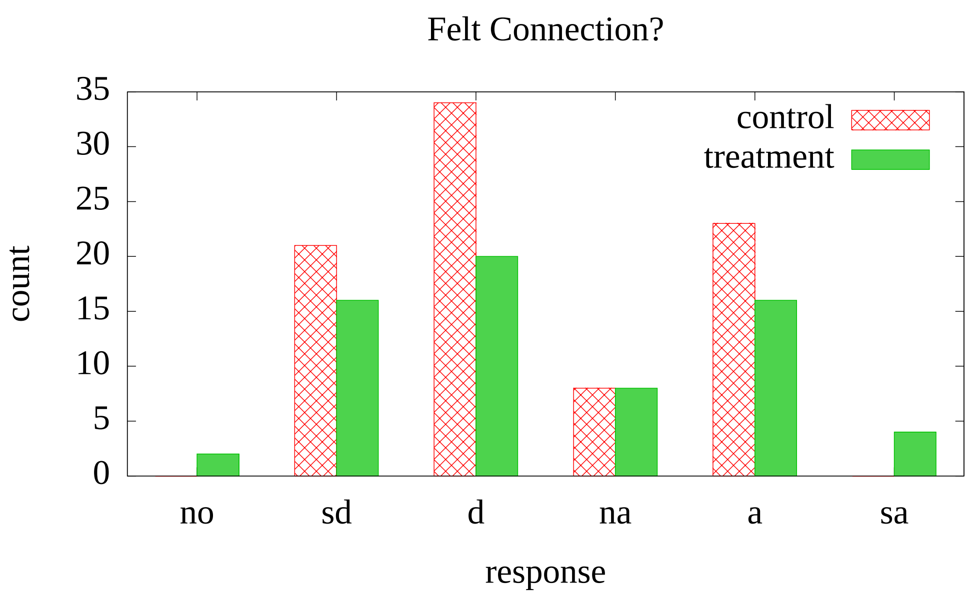


Figure 59: The distribution of responses to the Likert prompt: "I felt a sense of connection with the character in the story." There were 86 participants in the control group and 66 participants in the treatment group that answered this question.

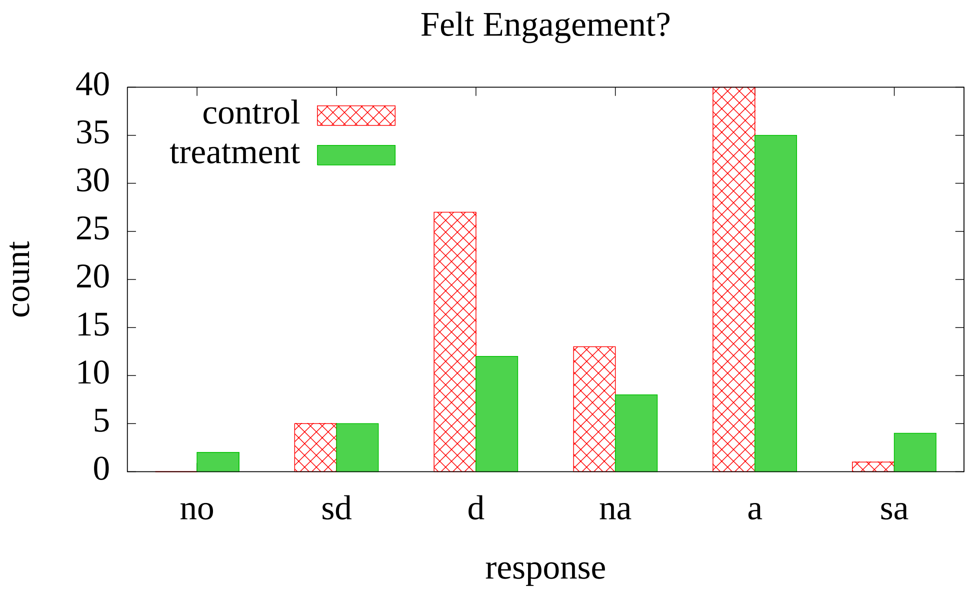


Figure 60: The distribution of responses to the Likert prompt: “I felt a sense of engagement with the system.” There were 86 participants in the control group and 66 participants in the treatment group that answered this question.

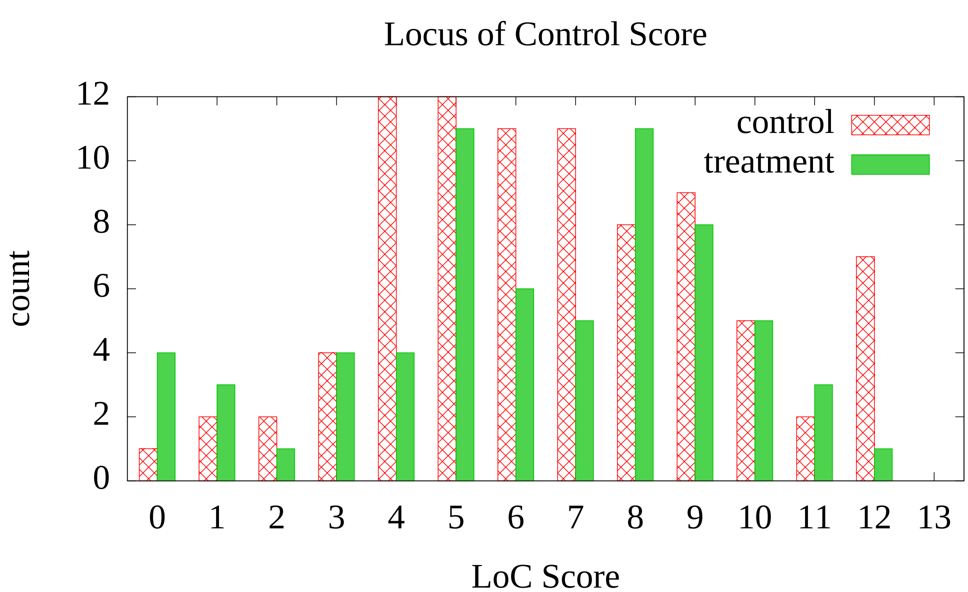


Figure 61: The distribution of Locus of Control scores. There were 75 participants in the control group and 66 participants in the treatment group that answered this question.

Table 25: A comparison of the predicted effectiveness of influence schema with observed effectiveness and re-computed effectiveness. The notation used here is introduced in Section 5.5. The first column indicates the depth of the story in which influence was applied. The second column contains the predicted effectiveness of the influence schema based on an assumption of $P(A_0 > C_0) = 0.5$. The third and fourth columns contain the observed probabilities of answer choices with and without the application of influence. The last column contains the predicted effectiveness of influence when the observed baseline is used as input. What is interesting to note is that at depth 1, where the effect of influence was found to be significant, our model's computed probability is fairly close.

Depth:	$P(A_1 > C_0)$ predicted	$P(A_0 > C_0)$ observerd	$P(A_1 > C_0)$ observed	$P(A_1 > C_0)$ computed
1:	0.7970	0.7065	**0.8514**	**0.9042**
2:	0.8300	0.2283	0.2568	0.5961
3:	0.7970	0.8369	0.7703	0.9527
4:	0.8300	0.2391	0.2297	0.6105
5:	0.8300	0.5152	0.2500	0.8413
	0.7970	0.4848	0.6111	0.7868
6:	0.8300	0.7500	1.0000	0.9374

APPENDIX E

A CASE STUDY OF AUTHORING WITH OUR TOOLS

For the implemented story used for the end-to-end evaluation presented in Chapter 7 we performed the duties of both *author* and *technologist*. As one of our main motivations for developing the tools and techniques described throughout this dissertation is to provide power for authors to create either more complex experiences or to create experiences of the same complexity as before only with a lesser effort, we devote this appendix to a recounting of the authorial effort we put forth to create the "The Side of Kobe Beef" interactive story. We describe the efforts for this story because we used all of the authoring tools presented in this dissertation to create that story experience. Here we will focus on the role of the author, separating out the tasks we preformed as the technologist. The reader interested in recreating our experiments or implementing our techniques for their own interactive story should hopefully find this appendix instructive. In addition, Appendix F contains the details of our story environment which may additionally help the reader wishing to recreate our experiments. Note that the entirety of Appendix F represents all of the authorial effort we put forth as well as a portion of the effort we made while acting as the technologist. At the end of this appendix, we will briefly discuss additional steps technologists working with authors (or authors working in a dual role as a technologist) can take to extend our implementation for use in other story environments. That description should help to characterize the parts of Appendix F that resulted from our efforts as the author and the parts that resulted from our efforts as the technologist.

In our role as the technologist, we have provided implementations of the key algorithms and corpora of schema and templates that provide the foundation upon which the drama manager for the interactive story is implemented. To leverage these tools, in our capacity as author we had to provide the following four components:

- the set of plot events that comprise the space of possible stories including the story text, information about the videos (if they are to be used), and the question-answer sets that drive the story forward as well as the precedence constraints and/or explicit transitions on those plot events
- the encoding of the player's likely behavior in the "base case" if no drama manager actions are applied
- the encoding of the author's desired aesthetics, in this case the target distribution
- the set of templates that serve as input into the schema provided by the technologist

In the following four sections, we will describe each of these processes further. In addition, we will provide screenshots of the authoring tool we implemented to aid in the authoring process.

E.1 The Story and Structure

The process of authoring the story can be approached in a number of ways. We chose to begin with a story that has already been published for two main reasons: 1) we are not authors by training; and 2) we wanted to try to control for any effects the story content may have on a player's exit survey responses. The use of an established story meant that we were tasked with converting a linear story into a branching story structure. To begin with we abstracted the linear story into a sequence of plot events and made the order explicit (*e.g.*, the graph of precedence constraints formed a chain).

Each story event represented a small piece of the larger story. Taken together, each of the events in the sequence comprised the entire story. To begin with, we associated text with each of the events. Note that the individual components of the story event can be reused across events—a feature we made extensive use of while converting a linear story

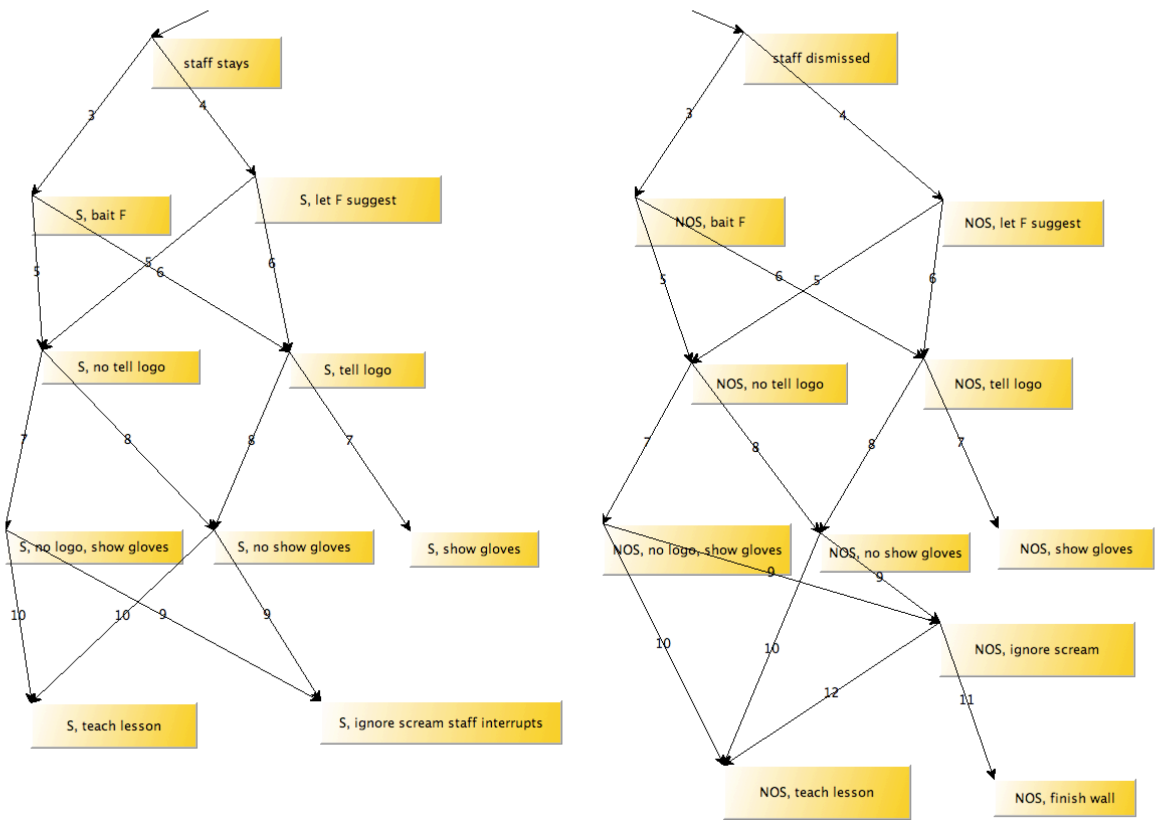


Figure 62: A screenshot of our graphical story editor in the structure editing phase. The author has the opportunity to put in transitions, providing a strict structure to the ordering of the story events.

to a branching story. To begin with, the linear structure of the base story is entered without any questions or answers as the linear structure does not lend itself to any choice points. Next, variants of the linear story are created. The process we followed involved assuming that a story event B that follows an earlier event A is triggered by answering a question that pertains to something specific about the contents of B. Thus, we could come up with an answer to the question to trigger B and a different answer to trigger an adaption of B, call it C, that is a modified version of B to reflect the different answer to the question. We followed this process for each of the events in the original linear story.

Figure 62 is a screenshot of our graphical editor in the structure editing phase. Once the story events have been entered into the system, the author needs to encode precedence constraints. These constraints can be extremely loose inducing a large set of valid paths

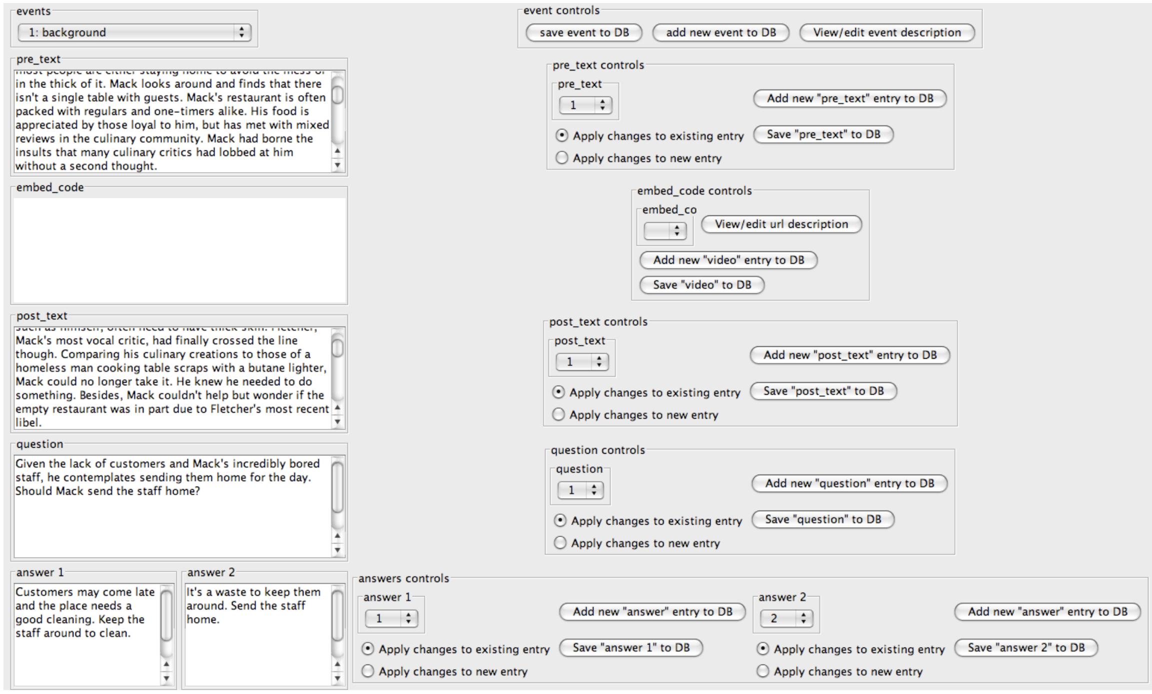


Figure 63: A screenshot of our graphical story editor in the event editing phase. The author has the opportunity to edit the text, the optional video information, and question-answer sets associated with the story event.

through the story space. On the other hand, these constraints can be very tight, functioning more like transitions than precedence constraints. Our graph of story events and precedence constraints was completely connected and the constraints on any vertex in the graph with multiple parents (*e.g.*, multiple precedence constraints) were implemented as "OR" type constraints, rather than "AND" type constraints. Thus, each precedence constraint could have an answer associated with it serving, in effect, as the transition from story event to story event.

One thing we encountered while authoring our story was the magnitude of the task of associating story text with the structured story events. It required some careful thought, but was not particularly time consuming to convert the linear story into the structured branching story we used; however, when it came time to write out the details of each of the story events, we found our authoring tool to be indispensable. Figure 63 shows one of the screens of our graphical story editor. Along the left side of the figure are six text entry areas corresponding to the "pre text", the embed code for the video (if used), the "post text", the question, and two answers. On the right side of the screen are a number of buttons and combo boxes that can be used to save individual portions of the story event to the database, associated entries from other story events with the current event, etc. Having this tool at our disposal made the process of authoring the text for story events and the structure of the story far more manageable.

To sum, the story authoring process required the creation of the story events, the associated story structure through the definition of precedence constraints, and the text to be presented to players.

E.2 The Player Model

The task of specifying the player model for our system is the task of estimating the probabilities that players make specific choices during the story when presented with questions. In the general case, what is required from the author is the definition of $P(t : e|a, t)$. That is, the probability that a particular story event e will occur immediately subsequent to a partial story t given that the drama manager has taken an action a. Note that one of the actions for which this model must be specified is the "no-op," where the drama manager chooses not to act ($P(t : e|\texttt{null}, t)$).

In earlier work on DODM and other drama management systems, the author would have to estimate the effectiveness of all possible drama manager actions in all possible story states. Fortunately, we have two distinct advantages in our setting. The first is question reuse and the second is our technique for domain transfer of player models (see Section 5.5). In the previous section, we described the process by which we generalized from the linear story we based our interactive story on. This generalization process led to significant reuse of question and answer sets. Specifically, every "depth" of the story had the same question and answer sets. So, while there were 34 distinct stories, there were only six different question and answer sets. As each answer was associated with a precedence constraint that functioned, for all intents and purposes, as a transition between story events, we only had to supply probability estimates for the 12 answers associated with the six questions. In other words, because the structure of our story was based on a linear story, we were able to make the assumption that the players' answers to questions would not be changed based on the story content up to that point. If an author creates a story structure where different branches of the story are vastly different, this assumption may not hold and the author would have to supply more probability estimates; however, in our story domain the structure enabled this simplification. Thus, because there were six forced-choice two-alternative questions and therefore six pairs of answers, we only had to supply six probability estimates and the other six were defined for us.

The use of our domain transfer model for estimating the effectiveness of drama manager actions meant that once the author provided "baseline" estimates of player choices for each of the questions, the technologist would provide the rest. These baseline estimates described the likelihood a player would choose one answer over another when no influence was applied. An author wishing to leverage our techniques and implement a player model need only provide these baseline estimates and then allow the technologist to do the rest.

E.3 The Aesthetics

The encoding of aesthetic goals for the drama manager to realize can be performed in a number of ways. In Section 3.4 we discussed a few variants of two main methods for this task: sampling using a reward function and using examples, or prototypes, and a distance measure. We used the second method to author our story. Thus, we will only briefly touch on the first method in this appendix.

The reward-sampling approach requires an author to specify a function that maps stories to a quality value. In general, this reward function can be any arbitrary function; typically in work on DODM the reward function is a linear combination of features about the story. The job of creating a target distribution from a reward function is one that would fall upon a technologist, not an author. That process is outlined in Section 3.4.

To author a target distribution directly using prototypes and distances requires the author to make three choices: 1) a set of prototypes and optionally prior weights for each; 2) a distance measure (or set of distance measures if the distribution is multivariate); and 3) a variance (or set of covariances if the distribution is multivariate). In the simplest case a standard distance measure like Levenshtein distance can be used. If the author wishes, they can create a distance measure that is more specific to their story domain. This process would be similar to the process of creating an evaluation function. In our case, we chose to use Levenshtein distance for its simplicity and because of the way in which our stories were authored; we did not have a need for a more complex distance measure. The next job for the author is to select prototypes. The author simply provides some examples of what they believe the good stories are. Optionally, the author can provide prior weights to these prototypes to indicate their relative preference for stories in the neighborhood of each of the prototypes. In our case, we opted to use a uniform prior, mainly due to the nature of our story environment. Lastly, the choice of variance is one the author should make in concert with the technologist (or on their own if a suitable authoring tool is available). The lower the variance, the higher the probability associated with the prototype stories. As the variance gets higher, more and more probability mass is attributed to stories farther and farther away from the prototypes. The only way to effectively select a variance is to plot the target distribution and visualize how quickly the target probability drops off as stories are farther and farther from the prototypes. In our case, we tested variances ranging from 0.25 to 2.0, but ultimately settled on 1.0.

E.4 The Input Templates

In order for an influence schema to be refined to a concrete action implementation, proper inputs must be available. In the case of our choose-your-own-adventure storytelling system, the refinement process results in text that can be displayed as part of a webpage. Thus, the

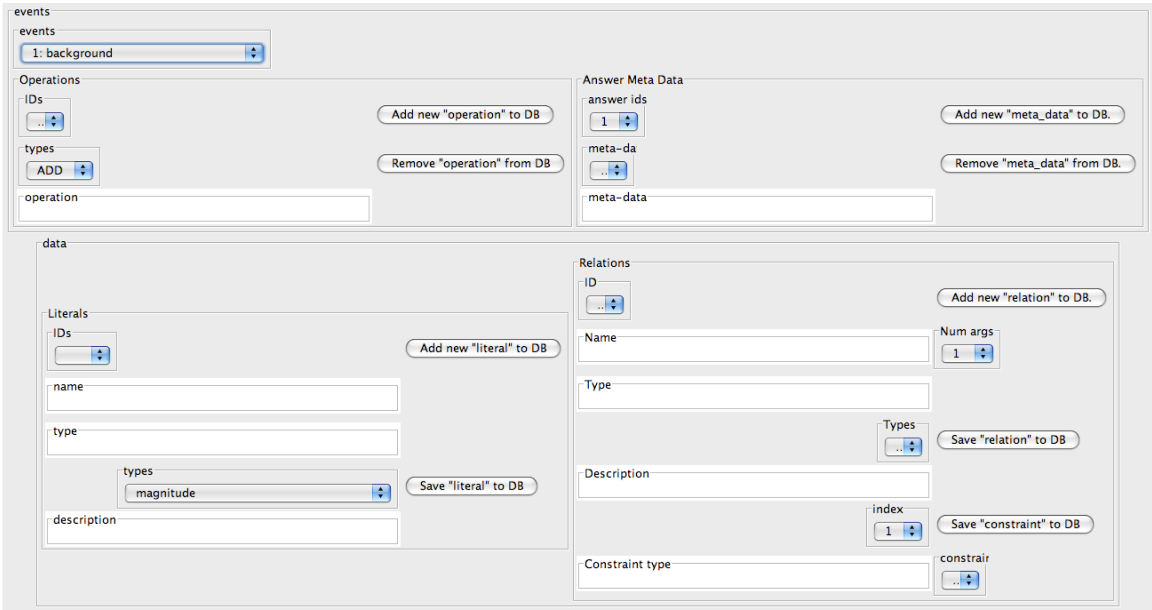


Figure 64: A screenshot of our graphical story editor in the template input phase. The author has the opportunity to input template names and to associate them with specific question answers.

author's job is to provide the natural language templates that are needed as input for each influence schema they wish to make available for the drama manager to apply to a particular answer. It is worth noting that process of specifying of input templates may differ for other storytelling environments. The determining factor for those differences is the technologist's solution to the action/plan refinement problem.

The authoring tool we implemented has a mode for input template specification to ease this process as well. The input template screen is shown in Figure 64. There are two steps to the process: creating the templates and associating the templates with an answer. Suppose the technologist has supplied a scarcity template that has three inputs: player, object, and verb. Further, suppose those inputs have type noun_proper, noun_singular, and verb_phrase respectively. In order to make that schema applicable to an answer, the author would need to provide templates containing the name of the player, an object associated with the answer, and a verb phrase describing the use of that object, each with the appropriate types to satisfy the schema inputs. To effectively provide the inputs, the author would need to inspect the schema to ensure they understand how each of the input templates are being used.

The author can provide enough inputs for more than one schema to be applicable to an answer choice. The solution to the TTD-SMDP tells the drama manager the relative frequency with which it should choose one of the available schema. In our case, we made at least one schema available for all of the 12 answers in the story definition. The majority had two or more possibilities.

It is worth noting that for the schema to refine properly, more than just the input templates need to be available for unification; however, each schema has a fixed set of templates necessary for unification that are independent of the inputs. When the technologist implements a new schema, they must also ensure that these "base" templates are available otherwise the schema will never be applicable. This is not, however, a job an author needs to perform unless they are filling both the author and technologist roles. In our dual role as author and technologist, we implemented a combined 53 templates, 29 of which were those used as inputs to schema and were specified in our role as author.

E.5 Tasks for Technologists

The four main tasks described above are the jobs the author needs to perform to create an interactive story in our environment. The technologist they work with provides some support for those tasks. In addition, the technologist provides the core algorithms needed for the drama manager to function. Specifically, the technologist provides an implementation of the TTD-SMDP, unification algorithm for action refinement, the set of influence schema, and the set of templates needed for the schema to unify. Further, the technologist can also provide the inputs for the domain transfer model that produces estimates of schema effectiveness.

Most of the algorithms, schema, and templates that we implemented for our second user study are applicable to other story domains. Indeed, one of the benefits to our approaches to these problems is that they are domain independent. The set of four authoring tasks described above is the minimum set of tasks an author must perform in order to use our techniques in other domains. Over time, as technologists contribute new schema and associated templates, corpora can be built up to enable more options for authors and drama managers when guiding players.

Should an author wish to use our techniques in another environment with characteristics different from our web-based storytelling system, the technologist will have to implement new solutions to many of the tasks associated with authoring. The influence schema are reusable across many types of domains, but the natural language templates and unification process as a solution to the refinement problem are not portable to other, potentially more complex, environments. Thus, the major job of the technologist when authoring for new types of environments is to define a new solution to the refinement problem. It is also worth noting that the author's fourth design problem of providing input templates may change depending on the technologist's solution to the refinement problem in different domains.

APPENDIX F

DETAILS OF “THE SIDE OF KOBE BEEF” STORY

In this appendix, we provide the details of “The Side of Kobe Beef” story used for our second user study, as well as the influence schema and natural language templates used for the generation and refinement processes. This appendix is essentially a “data dump” of all the components of the story. A reader wishing to duplicate our experiments could reproduce our interactive story and DODM instance with all of the data contained in this appendix. The details of the story itself are organized into a series of tables in the next section. Below that the six influence schemata are listed. Lastly, the 53 natural language templates are included as well.

F.1 The Story Details

Table 26 contains the event mapping, which lists all of the story events and their text components. The entries in that table refer to pre-text (Table 27), post-test (Table 28), questions (Table 29), and two answers (Table 30). Further, Table 31 contains the transitions which are a mapping from an event and answer combination, to a subsequent event. In Table 32 we list the names and types of the templates provided by the author that are input to the influence schema for unification (the templates themselves are listed in the final section of this appendix). Table 33 contains a list of the input templates that are associated with each answer in every event. At the end of this section is Table 34 which contains the two prototypes used to create the target distribution for the TTD-SMDP. With these prototypes, Levenshtein distance [62], and a variance of 1.0, the target distribution is fully specified.

Table 26: The mapping of pre_text, post_text, questions and answers for “The Side of Kobe Beef” story.

Event ID	Pre-text ID	Post-text ID	Question ID	Answer 1 ID	Answer 2 ID
1	1	1	1	1	2
2	2	2	2	3	4
3	3	3	2	3	4
4	4	4	3	5	6
5	5	13	3	5	6
6	4	4	3	5	6
7	5	13	3	5	6
8	6	5	4	7	8
9	7	5	4	7	8
10	6	5	4	7	8
11	7	5	4	7	8
12	8	6	0	0	0
13	9	7	5	9	10
14	10	7	5	9	10
15	8	6	0	0	0
16	9	7	5	9	10
17	10	7	5	9	10
18	11	8	6	11	12
19	13	10	0	0	0
20	14	11	0	0	0
21	13	10	0	0	0
22	12	9	0	0	0

Table 27: The pre-text entries for “The Side of Kobe Beef” story.

Pre Text ID	Pre Value
1	It is carnival season and the street outside Mack’s restaurant is empty. Not unusual for this time of year, most people are either staying home to avoid the mess or in the thick of it. Mack looks around and finds that there isn’t a single table with guests. Mack’s restaurant is often packed with regulars and one-timers alike. His food is appreciated by those loyal to him, but has met with mixed reviews in the culinary community. Mack had borne the insults that many culinary critics had lobbed at him without a second thought.
	Continued on next page

Table 27 — continued

Pre Text ID	Pre Value
2	Mack calls the staff together and instructs them to clean up and then go home. He grumbles to himself about the damage Fletcher has done to his business. There's no point in him keeping his staff around if there aren't any customers. Mack busied himself with some cleaning in the kitchen and managerial tasks. Before he knew it the staff had all gone home for the day.
3	Mack looks around helplessly at the bored staff. What if customers arrive later in the evening? He doesn't want to risk dismissing the staff early. He grumbles to himself about the damage Fletcher has done to his business. Mack busies himself with some cleaning in the kitchen and managerial tasks. Before he knows it, he is bored and without anything to do either.
4	Mack bides his time. For his plan to work, he needs the right opportunity. He dares not let slip his fury with Fletcher. As far as Fletcher is concerned, things are cordial at worst, even friendly between the two of them. Perhaps Fletcher assumes that Mack hasn't read his latest review. Mack, like many chefs, often doesn't read the critic's reviews of his restaurant. Why bother? The conversation continues for what seems like ages to Mack. Fletcher is happily vomiting the integral details of the fine cuisine he's had recently. And then, seemingly out of nowhere, Mack gets his opening. Fletcher is spewing words left and right, but Mack seizes on two of them "...Kobe beef...."
5	Mack bides his time. For his plan to work, he needs the right opportunity. He dares not let slip his fury with Fletcher. As far as Fletcher is concerned, things are cordial at worst, even friendly between the two of them. Perhaps Fletcher assumes that Mack hasn't read his latest review. Mack, like many chefs, often doesn't read the critic's reviews of his restaurant. Why bother? The conversation continues for what seems like ages to Mack. Finally, Mack takes matters in his own hands. Feigning a sudden realization, Mack exclaims "Kobe beef! I have come into possession of an entire side of Kobe beef. I can't be sure it is authentic though."
6	Mack, somewhat surprised by Fletcher's inquiry about the logo, stops to ponder a tactful response. Fletcher encourages him to answer. "It comes from my beliefs" Mack starts. Fletcher nods. "A wise man once said 'I must not only punish, but punish with impunity.' The depiction of a man, frozen in a block of ice up to his neck is a pretty serious punishment." Fletcher looks at Mack strangely. "Well, it's only a logo I guess. Let us see this side of Kobe beef!" he said as he gesticulated wildly in an attempt to spur Mack onward.

Continued on next page

Table 27 — continued

Pre Text ID	Pre Value
7	Mack, somewhat surprised by Fletcher's inquiry about the logo, stops to ponder a tactful response. Fletcher encourages him to answer. "It's personal" Mack starts. Fletcher nods. Mack thinks better of himself. He can't let Fletcher know how angry he is. "That's all you need to know" he continues. "Well, it's only a logo I guess. Let us see this side of Kobe beef!" Fletcher said as he gesticulated wildly in an attempt to spur Mack onward.
8	Mack gropes inside his pocket and grabs a hold of the heavy duty welder's gloves. As he begins to pull them out of his coat, he wonders what Fletcher will think when he sees them. Fletcher goads Mack, "Quit stalling man. The gloves. I want to see the beef." Mack obliges and hands him the gloves as they proceed through the kitchen. The freezer is located off the side of the reasonably-sized industrial kitchen. On either side of the large industrial doors are stainless steel counter tops used for butchery and other food prep. Below one of the counters are a few sheets of steel carefully, but not too carefully, placed around a Miller MIG welder.
9	Mack reaches his hand into his pocket and fondles the welder's gloves as they proceed toward the freezer. The freezer sits off of the main kitchen through two large industrial doors. The doors are skirted by large stainless steel counter tops on either side. Underneath one of the counter tops is a Miller MIG welding machine carefully, but not too carefully, surrounded by sheets of stainless steel. Mack glances at the welding machine as they walk nearer to the freezer entrance. Fletcher is no fool. If he spots the welder and sees the welder's gloves, he may begin to put things together. Mack decides to lie. "Sorry, no gloves. Surely someone of your knowledge can determine if it is Kobe beef without needing to handle it." Mack goads. "Of course I can!" replies Fletcher as he confidently enters the freezer. He scans the room, failing to see a side of beef anywhere. Mack motions him toward an opening in the wall leading to a back room. "The beef is in there?" Fletcher asks excitedly. Mack nods and follows him closely into the back room. It's dark and frigid in this cramped room. Fletcher puts his hands out to keep from walking face-first into the wall. When he reaches it, he turns to face Mack.

Continued on next page

Table 27 — continued

Pre Text ID	Pre Value
10	Mack gropes inside his pocket and grabs a hold of the heavy duty welder's gloves. As he begins to pull them out of his coat, he wonders what Fletcher will think when he sees them. Fletcher goads Mack, "Quit stalling man. The gloves. I want to see the beef." Mack obliges and hands him the gloves as they proceed through the kitchen. The freezer is located off the side of the reasonably-sized industrial kitchen. On either side of the large industrial doors are stainless steel counter tops used for butchery and other food prep. Below one of the counters are a few sheets of steel carefully, but not too carefully, placed around a Miller MIG welder. They enter the freezer, Fletcher in front of Mack. Fletcher scans the room, failing to see a side of beef anywhere. Mack motions him toward an opening in the wall leading to a back room. "The beef is in there?" Fletcher asks excitedly. Mack nods and follows him closely into the back room. It's dark and frigid in this cramped room. Fletcher puts his hands out to keep from walking face-first into the wall. When he reaches it, he turns to face Mack.
11	Mack sits back for a second and listens to the sound of Fletcher, now screaming, pleading to be let out of the freezer. "Alright, the joke is over." he says, "Now let me see the beef. The Kobe beef." Sparks fly and despite the noisy popping and hissing of the welding machine permanently affixing the new wall into place, Mack can hear Fletcher's words perfectly.
12	Mack snaps back to reality. "Heavy heart? It must be due to cold and dampness in freezer," he mutters under his breath. Aside from being skilled in the kitchen, Mack is also a skilled craftsman. He grew up working on steel sculptures with his father and his welding skills are superb. The wall he is constructing is flawless, indistinguishable from the other sections of the freezer. Mack works quickly and finishes the wall.
13	The screams emanating through the remaining opening in the wall are growing louder and more frantic by the minute. "Please, you can't do this Mack....the beef....let me out and I can help you identify it....PLEASE!" Mack's heavy heart weighs on him. After all, the sentiments behind his restaurant's logo were in fact just that...sentiments. Mack realizes that following through with the wall will fundamentally change him. This staunch realization is enough to bring Mack back to reality. Fletcher's torture can not continue. He has learned his lesson. Mack has no need to make things worse for him.

Continued on next page

Table 27 — continued

Pre Text ID	Pre Value
14	Mack sits back for a second and listens to the sound of Fletcher, now screaming, pleading to be let out of the freezer. “Alright, the joke is over.” he says, “Now let me see the beef. The Kobe beef.” Sparks fly and despite the noisy popping and hissing of the welding machine permanently affixing the new wall into place, Mack can hear Fletcher’s words perfectly.

Table 28: The post-text entries for “The Side of Kobe Beef” story.

Post Text ID	Post Value
1	The restaurant business is often cutthroat and chef’s, such as himself, often need to have thick skin. Fletcher, Mack’s most vocal critic, had finally crossed the line though. Comparing his culinary creations to those of a homeless man cooking table scraps with a butane lighter, Mack could no longer take it. He knew he needed to do something. Besides, Mack couldn’t help but wonder if the empty restaurant was in part due to Fletcher’s most recent libel.
2	The streets outside were still oddly empty. Mack sits at a table in the front sipping a glass of wine. A small figure appears off in the distance, but Mack doesn’t give it a second thought. Anger sears his veins. Mack was accustomed to shrugging off Fletcher’s insults, but things had gone too far. To attack a chef’s credibility is to attack his livelihood. The figure on the street grows as it approaches. But what can Mack do? What can a humble chef do when a critic rakes him over the coals? The figure has a face. Mack awkwardly gesticulates his surprise to see Fletcher walking down the street. He runs outside to greet him warmly.
3	The streets outside were still oddly empty. Mack sits at a table in the front sipping a glass of wine. Being the chef and restaurant owner has it’s perks, like drinking wine in front of the staff. A small figure appears off in the distance, but Mack doesn’t give it a second thought. Anger sears his veins. Mack was accustomed to shrugging off Fletcher’s insults, but things had gone too far. To attack a chef’s credibility is to attack his livelihood. The figure on the street grows as it approaches. But what can Mack do? What can a humble chef do when a critic rakes him over the coals? The figure has a face. Mack awkwardly gesticulates his surprise to see Fletcher walking down the street. He runs outside to greet him warmly.
	Continued on next page

Table 28 — continued

Post Text ID	Post Value
4	"Kobe beef!" Mack exclaims. "I have come into possession of an entire side of Kobe beef. I can't be sure it is authentic though." Fletcher can barely contain his excitement, "I can tell" he shouts. Mack knows he must be careful now. "It is hanging in my freezer, but you're a busy man. I'll ask Lester to help me. I don't want to impose." Fletcher laughs. "Lester can't tell Kobe beef from Wagyu, I'm the one you need. Let us go look now." Mack can't argue, so they proceed. As they enter the restaurant, Fletcher's eye fixes on the large circular logo behind the reception desk. He implores Mack to explain the meaning behind the logo.
5	The two men proceed through the dining room toward the back of the restaurant. "This way," Mack says, "the side is in the freezer. Delivered just today." Fletcher asks, "Is this real? A whole side of Kobe beef?" "I have my doubts, but I couldn't pass on the opportunity," Mack retorts. They round the corner and excitement is visible all over Fletcher's body. Mack is enjoying this immensely. "The beef!" says Fletcher, "I don't want to spoil it to examine it. Do you have gloves I can wear?"
6	Fletcher pulls the welder's gloves onto his hands and examines them quizzically. As he slowly lowers his hands to his sides, Fletcher's gaze fixes on the welding machine. Suddenly the look of puzzlement erodes from his face and his expression hardens. His eyes rapidly dart back and forth between the gloves and the welding machine. "What are you playing at?" he yells. Mack stutters. "Uh...uh...uh...THE BEEF!" he unintentionally yells. Mack looks at Fletcher to see the face of a man who is undoubtedly adding up the pieces. The frozen man in the logo representing Mack's beliefs about revenge, the steel, the welding machine, the glove. "Have you read my most recent reviews?" Fletcher asks nervously. Mack, trying but failing to sound confident replies, "Um, of course not. You know I never read reviews of my work." Fletcher sees through it. "You wouldn't know Kobe beef if it walked in and laid down on your plate!" he yells as he quickly runs out of the restaurant. His opporunity now lost, Mack hangs his head in sorrow. Fletcher will certainly destroy his reputation, and in doing so his business, as a result of this episode. Within two years, Mack is forced to close his restaurant.

Continued on next page

Table 28 — continued

Post Text ID	Post Value
7	As Fletcher turns, Mach thrusts him back against the wall and quickly wraps chains around his mid-section. He then locks the chains to the wall and takes a step back. "I don't see the beef." Fletcher says, "You said it was here." "Ah yes, the beef!" Mack replies as he walks out of the alcove and then the freezer. He bends down, grabs the stainless steel panels, and begins dragging them and the MIG welder toward the back of the freezer. Fletcher can be heard in the background quietly muttering to himself about the Kobe beef as Mack begins to weld a new wall into place enclosing Fletcher. Fletcher seems to have regained his composure now, and pleads to be shown the side of Kobe beef. Mack pauses to enjoy the sound of fear in Fletcher's voice.
8	"Ah yes, the Kobe beef" Mack says. "Yes, the beef!" exclaims Fletcher, "I can tell you for sure. You need me. Lester can't tell Kobe from Wagyu!" Mack laughs to himself as he continues to weld the wall into place. "The beef" he says. "Yes, the beef! Where is it?" Mack laughs again, but realizes his heart is heavy and he pauses to contemplate Fletcher's fate.
10	Mack retrieves a cutting wheel from the toolbox in the corner and begins to remove the wall he had almost finished erecting. Fletcher thanks him profusely as Mack's work nears completion. He crosses the room and removes the chains. Before allowing Fletcher to leave, Mack warns him against giving his restaurant a poor review again. In the weeks and years to come, business in Mack's restaurant flourishes.
9	Upon placing a shelf in front of the new wall, Mack removes any trace of anything ever having been behind it. He puts down his tools and listens to Fletcher's barely audible sobs emanating from behind the wall. Immurement is a certainty, especially at the freezing temperatures behind the wall. It shall be many a year before anyone notices anything. R.I.P. Fletcher!
11	Just as Mack is about to weld the final steel plate into place, he hears yelling. This isn't the sound of Fletcher's voice. This is coming from somewhere else. From behind him. Mack turns to find two of the restaurant staff leading a police officer, gun drawn, into the freezer. Mack has made a serious mistake in allowing his staff to remain while he sought revenge on Fletcher. It will be many years before Mack gets out of a real prison.

Continued on next page

Table 28 — continued

Post Text ID	Post Value
12	Fletcher can barely contain his excitement, "I can tell" he shouts. Mack knows he must be careful now. "It is hanging in my freezer, but you're a busy man. I'll ask Lester to help me. I don't want to impose." Fletcher laughs. "Lester can't tell Kobe beef from Wagyu, I'm the one you need. Let us go look now." Mack can't argue, so they proceed. As they enter the restaurant, Fletcher's eye fixes on the large circular logo behind the reception desk. He implores Mack to explain the meaning behind the logo.
13	Fletcher can barely contain his excitement, "I can tell" he shouts. Mack knows he must be careful now. "It is hanging in my freezer, but you're a busy man. I'll ask Lester to help me. I don't want to impose." Fletcher laughs. "Lester can't tell Kobe beef from Wagyu, I'm the one you need. Let us go look now." Mack can't argue, so they proceed. As they enter the restaurant, Fletcher's eye fixes on the large circular logo behind the reception desk. He implores Mack to explain the meaning behind the logo.

Table 29: The question entries for "The Side of Kobe Beef" story.

Question ID	Question Value
1	Given the lack of customers and Mack's incredibly bored staff, he contemplates sending them home for the day. Should Mack send the staff home?
2	Should Mack lure Fletcher back to the restaurant or let the conversation develop on it's own?
3	The logo is a large black circle with a crude depiction of a grown man frozen in a block of ice up to his neck. Should Mack explain the logo to Fletcher?
4	Mack has a set of welder's gloves in his coat. Should Mack offer the welder's gloves for Fletcher to wear or should he ignore the question and continue to the freezer?
5	Has Mack taught Fletcher a lesson? Or should Mack ignore Fletcher and continue working on the wall?
6	Should Mack end the joke and let Fletcher out of the room before sealing it?

Table 30: The answers used in "The Side of Kobe Beef."

Answer ID	Answer Value
1	"Customers may come late and the place needs a good cleaning. Keep the staff around to clean."
2	"It's a waste to keep them around. Send the staff home."
3	"Mack should lure Fletcher into his restaurant."
4	"Mack should ignore his anger and allow the conversation to develop organically."
5	"Mack should ignore the question and change the subject."
6	"Mack should tell Fletcher what the logo represents."
7	"Mack should give the welder's gloves to Fletcher."
8	"Mack should ignore the request for gloves and continue into the freezer."
9	"Mack still wishes to seek revenge, he should keep working on the wall."
10	"Mack has scared Fletcher sufficiently and should let him go."
11	"Mack should leave him there and finish the wall."
12	"Fletcher has learned not to mess with Mack and will not be insulting Mack's cooking any longer."

Table 31: The transition entries for "The Side of Kobe Beef" story mapping event ID and answer ID pairs to the next event.

Transition ID	From Event ID	Answer ID	Next Event ID
1	1	1	3
2	1	2	2
3	2	3	5
4	2	4	4
5	3	3	7
6	3	4	6
7	4	5	9
8	4	6	8
9	5	5	9
10	5	6	8
11	6	5	11
12	6	6	10
13	7	5	11
14	7	6	10
15	8	7	12

Continued on next page

Table 31 — continued

Transition ID	From Event ID	Answer ID	Next Event ID
16	8	8	13
17	9	7	14
18	9	8	13
19	10	7	15
20	10	8	16
21	11	7	17
22	11	8	16
23	13	9	18
24	13	10	19
25	14	9	18
26	14	10	19
27	16	9	20
28	16	10	21
29	17	9	20
30	17	10	21
31	18	11	22
32	18	12	19

Table 32: The template names and types used as input to the influence schema for unification.

Template ID	Template Name	Template Type
1	extremely	magnitude
2	very	magnitude
3	staff	noun_group
4	he	noun_proper
5	mack	noun_proper
6	the-conversation	noun_singular_event
7	the-freezer	noun_singular_event
8	the-gloves	noun_singular_event
9	the-logo	noun_singular_event
10	the-subject	noun_singular_event
11	the-taught-lesson	noun_singular_event
12	the-wall	noun_singular_event
13	carefully	phrase_rate
14	regularly	phrase_rate
15	few-minutes	phrase_rate|inflated_phrase_rate

Continued on next page

Table 32 — continued

Template ID	Template Name	Template Type
16	go_home_early	verb_phrase
17	lure-him	verb_phrase
18	benefit-from-lesson	verb_phrase|verb_phrase_trans
19	change-subject	verb_phrase|verb_phrase_trans
20	develop-conversation	verb_phrase|verb_phrase_trans
21	enter-freezer	verb_phrase|verb_phrase_trans
22	finish-wall	verb_phrase|verb_phrase_trans
23	give-gloves	verb_phrase|verb_phrase_trans
24	serve-customers	verb_phrase|verb_phrase_trans
25	staff-clean	verb_phrase|verb_phrase_trans
26	tell-logo	verb_phrase|verb_phrase_trans
27	cuban-cigar	noun_singular_item
28	fletcher	noun_proper_npc
29	drink-from-flask	verb_phrase_active|verb_phrase_current

Table 33: The mapping of input templates to answers and events.

Event ID	Answer ID	Input ID
1	1	5
1	1	4
1	1	25
1	1	24
1	1	14
1	1	1
1	2	5
1	2	4
1	2	3
1	2	16
2	3	5
2	3	4
2	3	17
2	3	15
2	4	5
2	4	4
2	4	20
2	4	15

Continued on next page

Table 33 — continued

Event ID	Answer ID	Input ID
2	4	1
2	4	6
3	3	5
3	3	4
3	3	17
3	3	15
3	4	5
3	4	4
3	4	20
3	4	15
3	4	1
3	4	6
4	5	5
4	5	4
4	5	19
4	5	10
4	6	5
4	6	26
4	6	13
4	6	1
4	6	9
5	5	5
5	5	4
5	5	19
5	5	10
5	6	5
5	6	26
5	6	13
5	6	1
5	6	9
6	5	5
6	5	4
6	5	19
6	5	10
6	6	5
6	6	26
6	6	13
6	6	1
6	6	9
7	5	5

Continued on next page

Table 33 — continued

Event ID	Answer ID	Input ID
7	5	4
7	5	19
7	5	10
7	6	5
7	6	26
7	6	13
7	6	1
7	6	9
8	7	5
8	7	23
8	7	13
8	7	2
8	7	8
8	8	5
8	8	4
8	8	21
8	8	7
8	8	15
9	7	5
9	7	23
9	7	13
9	7	2
9	7	8
9	8	5
9	8	4
9	8	21
9	8	7
9	8	15
10	7	5
10	7	23
10	7	13
10	7	2
10	7	8
10	8	5
10	8	4
10	8	21
10	8	7
10	8	15
11	7	5
11	7	23

Continued on next page

Table 33 — continued

Event ID	Answer ID	Input ID
11	7	13
11	7	2
11	7	8
11	8	5
11	8	4
11	8	21
11	8	7
11	8	15
13	9	5
13	9	4
13	9	15
13	10	5
13	10	4
13	10	18
13	10	13
13	10	2
13	10	11
14	9	5
14	9	4
14	9	15
14	10	5
14	10	4
14	10	18
14	10	13
14	10	2
14	10	11
16	9	5
16	9	4
16	9	15
16	10	5
16	10	4
16	10	18
16	10	13
16	10	2
16	10	11
17	9	5
17	9	4
17	9	15
17	10	5
17	10	4

Continued on next page

Table 33 — continued

Event ID	Answer ID	Input ID
17	10	18
17	10	13
17	10	2
17	10	11
18	11	5
18	11	4
18	11	22
18	11	12
18	11	15
18	12	5
18	12	4
18	12	18
18	12	13
18	12	2
18	12	11
4	6	27
4	6	28
5	6	27
5	6	28
6	6	27
6	6	28
7	6	27
7	6	28
11	7	28
11	7	29
10	7	28
10	7	29
9	7	28
9	7	29
8	7	28
8	7	29
13	9	22
13	9	12
14	9	22
14	9	12
16	9	22
16	9	12
17	9	22
17	9	12

Table 34: The prototypes for the target distribution.

Prototype ID	Event ID	Sequence Index
1	1	1
1	2	2
1	5	3
1	8	4
1	13	5
1	18	6
1	22	7
2	1	1
2	3	2
2	7	3
2	10	4
2	16	5
2	21	6

F.2 The Influence Schemata

The six influence schemata implemented for the end-to-end evaluation in study 2 are presented here. The goal probabilities were calculated according to the method outlined in Section 5.5 using a baseline assumption of $P(A_0 > C_0) = 0.5$. The input data for the first reciprocity schema was obtained from Cialdini's work on reciprocal concessions [28]. The input data for the second and third reciprocity schema came from Cialdini's book [26]. The input data for the three scarcity schemata was obtained from the results of study 1 [110].

Reciprocity Schema 1:

```
type: reciprocity
subtype: reciprocal_concessions
name: ask_then_retreat
comment: intended for use with opportunity
         to perform an action with some cost
goal_prob: 0.83

num_inputs: 4
num_actions: 2
```

```
input_1: <action>
input_2: <player>
input_3: <cost>
input_4: <high-cost>

input_type_1: verb_phrase
input_type_2: noun_proper
input_type_3: phrase_rate
input_type_4: inflated_phrase_rate

action_1: q:no,2:yes,0:obtain-commitment(
                         <player>,
                         apply(
                           <action>,
                           inflate(<high-cost>)
                         )
                       )
action_2: q:no,-1:yes,0:concede(
                          <player>,
                          at-cost(<high-cost>),
                          obtain-commitment(
                            <player>,
                            apply(<action>,<cost>)
                          )
                        )
```

Reciprocity Schema 2:

```
type: reciprocity
subtype: gift
name: unsolicited_favor_object
comment: works when another npc gives an object to the
         player
goal_prob: 0.71

num_inputs: 3
num_actions: 2

input_1: <player>
input_2: <npc>
input_3: <obj>

input_type_1: noun_proper
input_type_2: noun_proper_npc
input_type_3: noun_singular_item
```

```
action_1: unsolicited(hand-to(<npc>,<player>,<obj>))
action_2: speaks(<npc>,<player>,a-gift)
```

Reciprocity Schema 3:

```
type: reciprocity
subtype: gift
name: unsolicited_favor_action
comment: works when another npc performs an action for the
         player
goal_prob: 0.71

num_inputs: 4
num_preconds: 0
num_actions: 2
num_postconds: 0

input_1: <player>
input_2: <npc>
input_3: <action>
input_4: <action_2>

input_type_1: noun_proper
input_type_2: noun_proper_npc
input_type_3: verb_phrase_active
input_type_4: verb_phrase_current

action_1: sees(<player>,performing(<npc>,<action>))
action_2: offer(<npc>,<player>,<action_2>)
```

Scarcity Schema 1:

```
type: scarcity
subtype: reduce_time
name: reduce_supply_opportunities
comment: intended for use with opportunity to perform some
         action, rather than supply of an object
goal_prob: 0.797

num_inputs: 5
num_actions: 3

input_1: <player>
input_2: <quantity>
input_3: <value>
input_4: <verb>
input_5: <verbtrans>
```

```
input_type_1: noun_proper
input_type_2: phrase_rate
input_type_3: magnitude
input_type_4: verb_phrase
input_type_5: verb_phrase_trans

action_1: inform(<player>,needs(<verb>))
action_2: inform(<player>,has-value(<verbtrans>,<value>))
action_3: inform(<player>,is-reduced-time(<verbtrans>))
```

Scarcity Schema 2:

```
type: scarcity
subtype: impose_deadline
name: impose_deadline
comment: intended for use with opportunity to perform an
         action
goal_prob: 0.797

num_inputs: 3
num_actions: 2

input_1: <player>
input_2: <obj>
input_3: <verb>

input_type_1: noun_proper
input_type_2: noun_singular|noun_group
input_type_3: verb_phrase

action_1: inform(<player>,exists(<obj>))
action_2: inform(
            <player>,
            last-opportunity(exercise(<verb>,<obj>))
          )
```

Scarcity Schema 3:

```
type: scarcity
subtype: reduce_time
name: impose_deadline_2
comment: intended for use with opportunity to perform an
         action
goal_prob: 0.797

num_inputs: 3
```

```
num_actions: 2

input_1: <player>
input_2: <obj>
input_3: <verb>

input_type_1: noun_proper
input_type_2: noun_singular_event
input_type_3: verb_phrase

action_1: inform(<player>,exists(<obj>))
action_2: inform(<player>,soon-over(<verb>))
```

F.3 Natural Language Templates

Here we list the 53 natural language templates that made refinement of the influence schema possible. Of these 53 templates, 29 are the input templates listed in Table 32 above. The remaining were "base" templates that correspond to the hard-coded actions in the six influence schema listed in the previous section.

```
name: a-gift
type: phrase
num_params: 0
text: A gift for you
punctuation: !



name: apply
type: verb_clause
num_params: 2
param_name_1: <action>
param_name_2: <cost>
param_type_1: verb_phrase
param_type_2: phrase_rate
text: spend <cost> to <action>



name: at-cost
type: cost_phrase
```

```
num_params: 1
param_name_1: <cost>
param_type_1: inflated_phrase_rate
text: spend <cost>


name: benefit-from-lesson
type: verb_phrase|verb_phrase_trans
num_params: 0
text: benefit from teaching Fletcher a lesson
     |benefiting from having taught Fletcher a lesson


name: carefully
type: phrase_rate
num_params: 0
text: carefully


name: change-subject
type: verb_phrase|verb_phrase_trans
num_params: 0
text: change the subject|changing the subject


name: concede
type: sentence
num_params: 3
param_name_1: <player>
param_name_2: <cost>
param_name_3: <again>
param_type_1: noun_proper
param_type_2: cost_phrase
param_type_3: sentence
text: If <player> will not <cost>, <again>
punctuation: ?


name: cuban-cigar
type: noun_singular_item
num_params: 0
text: a hand-rolled cuban cigar
```

```
name: develop-conversation
type: verb_phrase|verb_phrase_trans
num_params: 0
text: allow the conversation to develop organically
     |allowing the conversation to develop organically
```

```
name: drink-from-flask
type: verb_phrase_active|verb_phrase_current
num_params: 0
text: taking a swig from a flask containing what smells like
          some very expensive bourbon
     |take a swig from the flask
```

```
name: enter-freezer
type: verb_phrase|verb_phrase_trans
num_params: 0
text: enter the freezer|entering the freezer
```

```
name: exercise
type: verb_phrase
num_params: 2
param_name_1: <phrase>
param_name_2: <obj>
param_type_1: verb_phrase
param_type_2: noun_group|noun_singular
text: let <obj> <phrase>
```

```
name: exists
type: noun_phrase
num_params: 1
param_name_1: <entity>
param_type_1: noun
             |noun_group
             |noun_singular
             |noun_singular_event
text: about <entity>
```

```
name: extremely
type: magnitude
num_params: 0
text: extremely



name: few-minutes
type: inflated_phrase_rate|phrase_rate
num_params: 0
text: half an hour|a few minutes



name: finish-wall
type: verb_phrase|verb_phrase_trans
num_params: 0
text: finish the wall|finishing the wall



name: fletcher
type: noun_proper_npc
num_params: 0
text: Fletcher



name: give-gloves
type: verb_phrase|verb_phrase_trans
num_params: 0
text: give the gloves to Fletcher|giving the gloves to
      Fletcher



name: hand-to
type: verb_phrase
num_params: 3
param_name_1: <npc>
param_name_2: <player>
param_name_3: <object>
param_type_1: noun_proper_npc
param_type_2: noun_proper
param_type_3: noun_singular_item
text: <npc> reaches into a pocket, produces <object>, and
      warmly hands it to <player>
```

```
name: has-quantity
type: noun_phrase
num_params: 2
param_name_1: <obj>
param_name_2: <quant>
param_type_1: verb_phrase_trans
param_type_2: phrase_rate
text: <obj> must be done <quant>


name: has-value
type: noun_phrase
num_params: 2
param_name_1: <obj>
param_name_2: <val>
param_type_1: verb_phrase_trans
param_type_2: magnitude
text: <obj> is <val> important


name: has-value-old
type: noun_phrase
num_params: 2
param_name_1: <obj>
param_name_2: <val>
param_type_1: verb_phrase_trans
param_type_2: magnitude
text: <obj> is of <val> value


name: he
type: noun_proper|noun_proper_object
num_params: 0
text: he|him


name: inflate
type: phrase_rate
num_params: 1
param_name_1: <act>
param_type_1: inflated_phrase_rate
text: <act>
```

```
name: inform
type: sentence
num_params: 2
param_name_1: <person>
param_name_2: <phrase>
param_type_1: noun_proper
param_type_2: noun_phrase|verb_phrase|phrase
text: <person> knows <phrase>


name: is-reduced-time
type: noun_phrase
num_params: 1
param_name_1: <action>
param_type_1: verb_phrase_trans
text: there will soon be no time for <action>


name: last-opportunity
type: noun_phrase
num_params: 1
param_name_1: <phrase>
param_type_1: verb_phrase
text: soon he will not be able to <phrase>


name: lure-him
type: verb_phrase
num_params: 0
text: lure Fletcher to the restaurant


name: mack
type: noun_proper|noun_proper_object
num_params: 0
text: Mack|Mack


name: needs
type: verb_phrase
num_params: 1
param_name_1: <action>
param_type_1: verb_phrase
text: he needs to <action>
```

```
name: no
type: answer
num_params: 0
text: No


name: obtain-commitment
type: sentence
num_params: 2
param_name_1: <player>
param_name_2: <phrase>
param_type_1: noun_proper_object
param_type_2: verb_clause
text: Is it worth it for <player> to <phrase>
punctuation: ?


name: offer
type: sentence
num_params: 3
param_name_1: <npc>
param_name_2: <player>
param_name_3: <action>
param_type_1: noun_proper_npc
param_type_2: noun_proper
param_type_3: verb_phrase_current
text: <npc> looks at <player>, then warmly offers for him to
          <action>


name: performing
type: verb_phrase_active
num_params: 2
param_name_1: <npc>
param_name_2: <action>
param_type_1: noun_proper_npc
param_type_2: verb_phrase_active
text: <npc> <action>


name: regularly
type: phrase_rate
num_params: 0
text: regularly
```

```
name: sees
type: sentence
num_params: 2
param_name_1: <player>
param_name_2: <action>
param_type_1: noun_proper
param_type_2: verb_phrase_active
text: Out of the corner of his eye, <player> sees <action>


name: go_home_early
type: verb_phrase
num_params: 0
text: go home early


name: serve-customers
type: verb_phrase|verb_phrase_trans
num_params: 0
text: have the staff serve customers|having the staff serve
         customers


name: soon-over
type: noun_phrase
num_params: 1
param_name_1: <phrase>
param_type_1: verb_phrase
text: soon he will not be able to <phrase>


name: speaks
type: sentence
num_params: 3
param_name_1: <npc>
param_name_2: <player>
param_name_3: <phrase>
param_type_1: noun_proper_npc
param_type_2: noun_proper
param_type_3: phrase
text: <npc> says to <player>, "<phrase>"
```

```
name: staff-clean
type: verb_phrase|verb_phrase_trans
num_params: 0
text: have the staff clean thoroughly|having the staff clean
         thoroughly


name: staff
type: noun_group
num_params: 0
text: the staff


name: tell-logo
type: verb_phrase|verb_phrase_trans
num_params: 0
text: describe the logo|describing the logo


name: the-conversation
type: noun_singular_event
num_params: 0
text: the conversation


name: the-freezer
type: noun_singular_event
num_params: 0
text: the freezer


name: the-gloves
type: noun_singular_event
num_params: 0
text: the gloves


name: the-logo
type: noun_singular_event
num_params: 0
text: the logo
```

```
name: the-subject
type: noun_singular_event
num_params: 0
text: the subject
```

```
name: the-taught-lesson
type: noun_singular_event
num_params: 0
text: the lesson he taught Fletcher
```

```
name: the-wall
type: noun_singular_event
num_params: 0
text: the wall
```

```
name: unsolicited
type: sentence
num_params: 1
param_name_1: <action>
param_type_1: verb_phrase
text: Without warning, <action>
```

```
name: very
type: magnitude
num_params: 0
text: very
```

```
name: yes
type: answer
num_params: 0
text: Yes
```

REFERENCES

[1] Andrade, G., Ramalho, G., Gomes, A. S., and Corruble, V., "Challenge-Sensitive Action Selection: an Application to Game Balancing," in *Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT05)*, pp. 194–200, IEEE Computer Society, 2005.

[2] Andrade, G., Ramalho, G., Gomes, A. S., and Corruble, V., "Challenge-Sensitive Game Balancing: an Evaluation of User Satisfaction," in *Proceedings of the Fourth Brazilian Workshop on Computer Games and Digital Entertainment (WJogos05)*, pp. 66–76, 2005.

[3] Andrade, G., Ramalho, G., Gomes, A. S., and Corruble, V., "Extending Reinforcement Learning to Provide Dynamic Game Balancing," in *Proceedings of the 2005 IJCAI Workshop on Reasoning, Representation, and Learning in Computer Games* (Aha, D. W., Muoz-Avila, H., and van Lent, M., eds.), pp. 7–12, 2005.

[4] Andrade, G., Ramalho, G., Gomes, A. S., and Corruble, V., "Dynamic Game Balancing: an Evaluation of User Satisfaction," in *Proceedings of the Second Artificial Intelligence and Interactive Digital Entertainment Conference (AIIDE06)*, AAAI Press, 2006.

[5] Andrade, G., Santana, H., Furtado, A., Leito, A., and Ramalho, G., "Online Adaptation of Computer Games Agents: A Reinforcement Learning Approach," *Scientia*, vol. 15, no. 2, 2004.

[6] Ariely, D., *Predictably Irrational: The Hidden Forces That Shape Our Decisions*. Harper Collins, 2008.

[7] Arinbjarnar, M., Barber, H., and Kudenko, D., "A critical review of interactive drama systems," in *AISB 2009 Convention: Adaptive & Emergent Behavior & Complex System, Symposium on AI & Games*, 2009.

[8] Asimov, I., "The miss america pageant," in *TV Guide*, August 30 1975.

[9] Bandura, A., Grusec, J. E., and Menlove, F. L., "Vicarious extinction of avoidance behavior," *Journal of Personality and Social Psychology*, vol. 5, pp. 16–23, 1967.

[10] Bandura, A. and Menlove, F. L., "Factors determining vicarious extinction of avoidance behavior through symbolic modeling," *Journal of Personality and Social Psychology*, vol. 8, pp. 99–108, 1968.

[11] BARBER, H. and KUDENKO, D., "A User Model for the Generation of Dilemma-based Interactive Narratives," in *Proceedings of the Second Workshop on Optimizing Player Satisfaction*, 2007.

[12] BARBER, H. and KUDENKO, D., "Dynamic Generation of Dilemma-based Interactive Narratives," in *Proceedings of the Third Artificial Intelligence and Interactive Digital Entertainment conference (AIIDE07)*, 2007.

[13] BATES, J., "Virtual Reality, Art, and Entertainment," *Presence: The Journal of Teleoperators and Virtual Environments*, vol. 2, pp. 133–138, 1992.

[14] BAUER, J. and LEVY, M., *How to Persuade People Who Don't Want to be Persuaded.* Wiley, 2004.

[15] BHAT, S., ROBERTS, D. L., NELSON, M. J., ISBELL, C. L., and MATEAS, M., "A Globally Optimal Algorithm for TTD-MDPs," in *Proceedings of the Sixth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS07)*, 2007.

[16] BOYD, S. and VANDENBERGHE, L., *Convex Optimization*. Cambridge University Press, 2004.

[17] BROEDER, D. W., "The university of chicago jury project," *Nebraska Law Review*, vol. 38, pp. 760–774, 1959.

[18] BRUCKMAN, A., "The combinatorics of storytelling: Mystery train interactive," 1990. Interactive Cinema Group internal paper, MIT Media Lab (Available by contacting the Interactive Cinema Group of the MIT Media Lab at the author's address.).

[19] BURGESS, R. and TURNER, S., "Seven key features for creating and sustaining commitment," *International Journal of Project Management*, vol. 18, no. 4, pp. 225–233, 2000.

[20] BUSEMANN, S. and HORACEK, H., "A flexible shallow approach to text generation," in *Proceedings of the Ninth International Workshop on Natural Language Generation*, pp. 238–247, 1998.

[21] CANTINO, A. S., ROBERTS, D. L., and ISBELL, C. L., "Autonomous Nondeterministic Tour Guides: Improving Quality of Experience with TTD-MDPs," in *Proceedings of the Sixth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS07)*, 2007.

[22] CANTINO, A. S., ROBERTS, D. L., and ISBELL, C. L., "Autonomous Nondeterministic Tour Guides: Improving Quality of Experience with TTD-MDPs," Tech. Rep. GIT-IIC-07-02, School of Interactive Computing, College of Computing, Georgia Institute of Technology, 2007.

[23] CHEN, S., NELSON, M., and MATEAS, M., "Evaluating the Authorial Leverage of Drama Management," in *Proceedings of the 2009 AAAI Spring Symposium on Intelligent Narrative Technologies II*, 2009.

[24] CHRISTIAN, D. and YOUNG, R. M., "Comparing Cognitive and Computational Models of Narrative Structure," Tech. Rep. TR03-001, Liquid Narrative Group, Department of Computer Science, North Carolina State University, 2003.

[25] CHURCH, D., "Formal abstract design tools," *Game Developer Magazine*, 1999.

[26] CIALDINI, R. B., *Influence: The Psychology of Persuasion*. Collins, 1998.

[27] CIALDINI, R. B., WOSINSKA, W., BARRETT, D. W., BUTNER, J., and GORNIK-DUROSE, M., "Compliance with a request in two cultures: The differential influence of social proof and commitment/consistency on collectivists and individualists," *Personality and Social Psychology Bulletin*, vol. 25, no. 10, pp. 1242–1253, 1999.

[28] CIALDINI, R. B., VINCENT, J. E., LEWIS, S. K., CATALAN, J., WHEELER, D., and DARBY, B. L., "Reciprocal concessions procedure for inducing compliance: The door-in-the-face technique," *Journal of Personality and Social Psychology*, vol. 31, no. 2, pp. 206–215, 1975.

[29] CRAWFORD, C., *Chris Crawford on Interactive Storytelling*. New Riders Games, 2004.

[30] DANIEL J. HOWARD, C. G. and JAIN, A., "What's in a name? a complimentary means of persuasion," *The Journal of Consumer Research*, vol. 22, no. 2, pp. 200–211, 1995.

[31] DEMASI, P. and CRUZ, A., "Online Coevolution for Action Games," in *Proceedings of The Third International Conference on Intelligent Games And Simulation*, pp. 113–120, 2002.

[32] DIEKMANN, A., "The power of reciprocity," *Journal of Conflict Resolution*, vol. 48, no. 4, pp. 487–505, 2004.

[33] DOW, S., *Understanding User Engagement in Immersive and Interactive Stories*. PhD thesis, School of Interactive Computing, Georgia Institute of Technology, 2008.

[34] DRACHMAN, D., DECARUFEL, A., and INKSO, C. A., "The extra credit effect in interpersonal attraction," *Journal of Experimental Social Psychology*, vol. 14, pp. 458–467, 1978.

[35] EAGLY, A. H., ASHMORE, R. D., MAKHIJANI, M. G., and LONGO, L. C., "What is beautiful is good, but...: A meta-analytic review of research on the physical attractiveness stereotype," *Psychological Bulletin*, vol. 110, no. 1, pp. 109–128, 1991.

[36] FAIRCLOUGH, C., *Story Games and the OPIATE System*. PhD thesis, University of Dublin - Trinity College, 2006.

[37] FALMAGNE, J. C., "Foundations of fechnerian psychophysics," in *Contemporary Developments in Mathematical Psychology* (KRANTZ, D. H., ATKINSON, R. C., LUCE, R. D., and SUPPES, P., eds.), Freeman, 1974.

[38] FLEISS, J. L., LEVIN, B., and PAIK, M. C., *Statistical Methods for Rates and Proportions*. John Wiley & Sons, 3 ed., 2003.

[39] FOLKES, V. S., MARTIN, I. M., and GUPTA, K., "When to say when: Effects of supply on usage," *Journal of Consumer Research*, vol. 20, pp. 467–477, 1993.

[40] FROMKIN, H. L. and BROCK, T. C., "A commodity theory analysis of persuasion," *Representative Research in Social Psychology*, vol. 2, no. 2, pp. 47–57, 1971.

[41] GELFOND, M. and LIFSCHITZ, V., "Representing action and change by logic programs," *The Journal of Logic Programming*, vol. 17, no. 2–4, pp. 301–321, 1993.

[42] GOLDSTEIN, N. J., MARTIN, S. J., and CIALDINI, R. B., *Yes! 50 Scientifically Proven Ways to Be Persuasive*. Free Press, 2008.

[43] GUILFORD, J. P., *Psychometric Methods*. Macmillan, 2nd ed., 1954.

[44] HAMMING, R. W., "Error-detecting and error-correcting codes," *Bell System Technical Journal*, vol. 29, no. 2, pp. 147–160, 1950.

[45] HARRELL, D. F. and ZHU, J., "Agency play: Dimensions of agency for interactive narrative design," in *Proceedings of the AAAI 2009 Spring Symposium on Interactive Narrative Technologies II*, pp. 156–162, 2009.

[46] HASTINGS, W. K., "Monte carlo sampling methods using markov chains and their applications," *Biometrika*, vol. 57, no. 1, pp. 97–109, 1970.

[47] HOFLING, C. K., BROTZMAN, E., DALRYMPLE, S., GRAVES, N., and PIERCE, C. M., "An experimental study in nurse-physician relationships," *The Journal of Nervous and Mental Disease*, vol. 143, no. 2, pp. 171–180, 1966.

[48] HOVY, E. H., "Automated discourse generation using discourse structure relations," *Artificial Intelligence (Special Issue on Natural Language Processing)*, vol. 63, 1993.

[49] HOWARD, R. A., *Dynamic Programming and Markov Processes*. Technology Press-Wiley, 1960.

[50] HOWARD, R. A., *Dynamic Probabilistic Systems, volume II*. John Wiley & Sons, 1971.

[51] HUNICKE, R. and CHAPMAN, V., "AI for Dynamic Difficulty Adjustment in Games," in *Challenges in Game Artificial Intelligence AAAI Workshop*, pp. 91–96, 2004.

[52] ISBELL, C. L., SHELTON, C. R., KEARNS, M., SINGH, S., and STONE, P., "A social reinforcement learning agent," in *Proceedings of the Fifth International Conference on Autonomous Agents (Agents01)*, pp. 377–384, 2001.

[53] KAELBLING, L. P., LITTMAN, M. L., and MOORE, A. P., "Reinforcement learning: A survey," *Journal of Artificial Intelligence Research*, vol. 4, pp. 237–285, 1996.

[54] KAELBLING, L., LITTMAN, M., and CASSANDRA, A., "Planning and acting in partially observable stochastic domains," *Artificial Intelligence Journal*, vol. 101, pp. 99–134, 1998.

[55] KNISHINSKY, A., *The Effects of Scarcity of Material and Exclusivity of Information on Industrial Buyer Perceived Risk in Provoking a Purchase Decision*. PhD thesis, Arizona State University, 1982.

[56] LAIRD, J. E., NEWELL, A., and ROSENBLOOM, P. S., "SOAR: An Architecture for General Intelligence," *Artificial Intelligence*, vol. 33, no. 3, pp. 1–64, 1987.

[57] LAKHANI, D., *Persuasion: The Art of Getting What You Want*. Wiley, 2005.

[58] LAMBERT, L. and CARBERRY, S., "A tripartite plan-based model of dialogue," in *Proceedings of the 29th annual meeting on Association for Computational Linguistics*, pp. 47–54, 1991.

[59] LAMSTEIN, A. and MATEAS, M., "A Search-Based Drama Manager," in *Proceedings of the AAAI-04 Workshop on Challenges in Game AI*, 2004.

[60] LAUREL, B., *Toward the Design of a Computer-Based Interactive Fantasy System*. PhD thesis, Drama department, Ohio State University, 1986.

[61] LEIBLER, S. K. R. A., "On information and sufficiency," *Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[62] LEVENSHTEIN, V. I., "Binary codes capable of correcting deletions, insertions and reversals," *Soviet Physics Doklady*, vol. 10, no. 8, pp. 707–710, 1966.

[63] LIKERT, R., "A technique for the measurement of attitudes," *Archives of Psychology*, vol. 140, pp. 1–55, 1932.

[64] LITTMAN, M. L., "Markov games as a framework for multi-agent reinforcement learning," in *Proceedings of the Eleventh International Conference on Machine Learning (ICML94)*, pp. 157–163, 1994.

[65] LUCE, R. D., *Individual Choice Behavior*. Wiley, 1959.

[66] LUCE, R. D. and GALANTER, E., "Discrimination," in *Handbook of Mathematical Psychology* (LUCE, R. D., BUSH, R. R., and GALANTER, E., eds.), vol. 1, pp. 191–243, 1963.

[67] MAGERKO, B., "Story Representation and Interactive Drama," in *Proceedings of the First Artificial Intelligence and Interactive Digital Entertainment Conference (AIIDE05)*, 2005.

[68] MAGERKO, B., *Player Modeling in the Interactive Drama Architecture*. PhD thesis, Department of Computer Science and Engineering, University of Michigan, 2006.

[69] MAGERKO, B. and LAIRD, J. E., "Building an Interactive Drama Architecture," in *Proceedings of the First International Conference on Technologies for Interactive Digital Storytelling and Entertainment (TIDSE03)*, pp. 226–237, 2003.

[70] MAGERKO, B. and LAIRD, J. E., "Mediating the Tension Between Plot and Interaction," in *Proceedings of the AAAI Workshop Series: Challenges in Game Artificial Intelligence*, pp. 108–112, 2004.

[71] MAGERKO, B., LAIRD, J. E., ASSANIE, M., KERFOOT, A., and STOKES, D., "AI Characters and Directors for Interactive Computer Games," in *Proccedings of the 16th Innovative Applications of Artificial Intelligence Conference (IAAI04)*, pp. 877–883, 2004.

[72] MANN, W. C. and THOMPSON, S. A., "Rhetorical structure theory: Toward a functional theory of text organization," *Text-Interdisciplinary Journal for the Study of Discourse*, vol. 8, no. 3, pp. 243–281, 1988.

[73] MATEAS, M. and STERN, A., "Structuring Content in the Façade Interactive Drama Architecture," in *Proceedings of the First Artificial Intelligence and Interactive Digital Entertainment (AIIDE05)*, 2005.

[74] MATEAS, M., "An Oz-Centric Review of Interactive Drama and Believable Agents," in *AI Today: Recent Trends and Developments* (WOODRIDGE, M. and VELOSO, M. M., eds.), vol. 1600 of *Lecture Notes in AI*, Springer, 1999.

[75] MATEAS, M., "A preliminary poetics for interactive drama and games," *Digital Creativity*, vol. 12, no. 3, pp. 140–152, 2001.

[76] MATEAS, M., *Interactive drama, art and artificial intelligence*. PhD thesis, School of Computer Science, 2002.

[77] MATEAS, M. and STERN, A., "Integrating Plot, Character, and Natural Language Processing in the Interactive Drama Façade," in *Proceedings of the First International Conference on Technologies for Interactive Digital Storytelling and Entertainment (TIDSE03)*, 2003.

[78] MATEAS, M. and STERN, A., "Towards Integrating Plot and Character for Interactive Drama," in *Working Notes of the Social Intelligent Agents: The Human in the Loop Symposium, AAAI Fall Symposium Series*, 2000.

[79] MATEAS, M. and STERN, A., "Architecture, Authorial Idioms and Early Observations of the Interactive Drama Façade," Tech. Rep. CMU-CS-02-198, Carnegie Mellon Univeristy, 2003.

[80] MATEAS, M. and STERN, A., "Façade: An Experiment in Building a Fully-Realized Interactive Drama," in *Game Developers Conference: Game Design Track*, 2003.

[81] MEDLER, B. and MAGERKO, B., "Scribe: A general tool for authoring interactive drama," in *Third International Conference on Technologies for Interactive Digital Storytelling and Entertainment*, pp. 139–150, 2006.

[82] METROPOLIS, N., ROSENBLUTH, A. W., ROSENBLUTH, M. N., TELLER, A. H., and TELLER, E., "Equations of state calculations by fast computing machines," *Journal of Chemical Physics*, vol. 21, no. 6, pp. 1087–1092, 1953.

[83] MEYEROWITZ, B. E. and CHAIKEN, S., "The effect of message framing on breast self-examination attitudes, intentions, and behavior," *Journal of Personality and Social Psychology*, vol. 52, no. 3, pp. 500–510, 1987.

[84] MILGRAM, S., *Obedience to Authority: An Experimental View*. Hapercollins, 1974.

[85] MOORE, J. D. and PARIS, C. L., "Planning text for advisory dialogues: Capturing intentional and rhetorical information," *Computational Linguistics*, vol. 19, pp. 651–694, 1993.

[86] MORIARTY, T., "Crime, commitment, and the responsive bystander," *Journal of Personality and Social Psychology*, vol. 31, pp. 370–376, 1975.

[87] MOTT, B. W., CALLAWAY, C. B., ZETTLEMOYER, L. S., LEE, S., and LESTER, J. C., "Towards Narrative-Centered Learning Environments," in *Proceedings of the AAAI Fall Symposium on Narrative Intelligence*, pp. 78–82, 1999.

[88] MOTT, B. W., LEE, S., and LESTER, J. C., "Probabilistic Goal Recognition in Interactive Narrative Environments," in *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI06)*, 2006.

[89] MOTT, B. W. and LESTER, J. C., "Narrative-Centered Tutorial Planning for Inquiry-Based Learning Environments," in *Proceedings of the Eigth International Conference on Intelligent Tutoring Systems (ITS06)*, 2006.

[90] MOTT, B. W. and LESTER, J. C., "U-Director: A Decision-Theoretic Narrative Planning Architecture for Storytelling Environments," in *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS06)*, pp. 977–984, ACM Press, 2006.

[91] MOTT, B. W., MCQUIGGAN, S. W., LEE, S., LEE, S., and LESTER, J. C., "Narrative-Centered Environments for Guided Exploratory Learning," in *Proceedings of the Agent Based Systems for Human Learning Workshop at the 5th International Joint Conference on Autonomous Agents and Multiagent Systems (ABSHLE06)*, 2006.

[92] MURRAY, J., *Hamlet on the Holodeck*. Free Press, 1997.

[93] NELSON, M. J. and MATEAS, M., "Search-Based Drama Management in the Interactive Fiction Anchorhead," in *Proceedings of the First Artificial Intelligence and Interactive Digital Entertainment Conference (AIIDE05)*, 2005.

[94] NELSON, M. J. and MATEAS, M., "Another look at search-based drama management," in *Proceedings of the 23rd AAAI Conference on Artificial Intelligence*, 2008.

[95] NELSON, M. J., MATEAS, M., ROBERTS, D. L., and ISBELL, C. L., "Declarative Optimization-Based Drama Management in the Interactive Fiction Anchorhead," *IEEE Computer Graphics and Applications (Special Issue on Interactive Narrative)*, vol. 26, no. 3, pp. 30–39, 2006.

[96] NELSON, M. J., ROBERTS, D. L., ISBELL, C. L., and MATEAS, M., "Reinforcement Learning for Declarative Optimization-Based Drama Management," in *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS06)*, 2006.

[97] ONTAÑÓN, S., JAIN, A., MEHTA, M., and RAM, A., "Developing a drama management architecture for interactive fiction games," in *Proceedings of the First Joint International Conference on Interactive Digital Storytelling (ICIDS-08)*, 2008.

[98] PETROVA, P. K., CIALDINI, R. B., and SILLS, S. J., "Consistency-based compliance across cultures," *Journal of Experimental Social Psychology*, vol. 43, no. 1, pp. 104–111, 2007.

[99] REEVES, B. and NASS, C., *The media equation: How people treat computers, television, and new media like real people and places*. Cambridge University Press, 1996.

[100] REGAN, D. T., "Effects of a favor and liking on compliance," *Journal of Experimental Social Psychology*, vol. 7, pp. 627–639, 1971.

[101] REGAN, D. T., "Effects of a favor and liking on compliance," *Journal of Experimental Social Psychology*, vol. 7, pp. 637–639, 1971.

[102] RIEDL, M. O., SARETTO, C. J., and YOUNG, R. M., "Managing Interaction Between Users and Agents in a Multi-Agent Storytelling Environment," in *Proceedings of the Second International Joint Conference on Autonomous Agents and Multi Agent Systems (AAMAS03)*, 2003.

[103] RIEDL, M. O. and STERN, A., "Believable Agents and Intelligent Scenario Direction for Social and Cultural Leadership Training," in *In Proceedings of the 15th Conference on Behavior Representation in Modeling and Simulation (BRIMS06)*, 2006.

[104] RIEDL, M. O. and STERN, A., "Believable Agents and Intelligent Story Adaptation for Interactive Storytelling," in *In Proceedings of the Third International Conference on Technologies for Interactive Digital Storytelling and Entertainment (TIDSE06)*, 2006.

[105] RIEDL, M. O. and STERN, A., "Failing Believably: Toward Drama Management with Autonomous Actors in Interactive Narratives," in *Proceedings of the Third International Conference on Technologies for Interactive Digital Storytelling and Entertainment (TIDSE06)*, 2006.

[106] RIEDL, M. O., STERN, A., DINI, D., and ALDERMAN, J., "Dynamic experience management in virtual worlds for entertainment, education, and training," *International Transactions on Systems Science and Applications, Special Issue on Agent Based Systems for Human Learning*, vol. 4, no. 2, 2008.

[107] RIEDL, M. O. and YOUNG, R. M., "From Linear Story Generation to Branching Story Graphs," *IEEE Computer Graphics and Applications (Special Issue on Interactive Narrative)*, vol. 26, no. 3, 2006.

[108] ROBERTS, D. L., BHAT, S., CLAIR, K. S., and ISBELL, C. L., "Authorial Idioms for Target Distributions in TTD-MDPs," in *Proceedings of the 22nd Conference on Artificial Intelligence (AAAI07)*, 2007.

[109] ROBERTS, D. L., CANTINO, A. S., and ISBELL, C. L., "Player Autonomy Versus Designer Intent: A Case Study of Interactive Tour Guides," in *Proceedings of the Third Conference and Artificial Intelligence for Interactive Digital Entertainment (AIIDE07)*, 2007.

[110] ROBERTS, D. L., FURST, M. L., ISBELL, C. L., and DORN, B., "Using Influence and Persuasion to Shape Player Experiences," in *In 2009 Sandbox: ACM SIGGRAPH Video Game Proceedings (SIGGRAPH Sandbox)*, 2009.

[111] ROBERTS, D. L. and ISBELL, C. L., "A survey and qualitative analysis of recent advances in drama management," *International Transactions on Systems Science and Applications, Special Issue on Agent Based Systems for Human Learning*, vol. 3, no. 1, pp. 61–75, 2008.

[112] ROBERTS, D. L., NELSON, M. J., ISBELL, C. L., MATEAS, M., and LITTMAN, M. L., "Targeting Specific Distributions of Trajectories in MDPs," in *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI06)*, 2006.

[113] ROBERTS, D. L., RIEDL, M. O., and ISBELL, C. L., "Beyond adversarial: The case of game ai as storytelling," in *Breaking New Ground: Innovation in Games,*

Play, Practice and Theory, Proceedings of the Digital Games Research Association 2009 Conference (DIGRA 2009), 2009.

[114] ROBERTS, D. L., STRONG, C. R., and ISBELL, C. L., "Estimating player satisfaction through the author's eyes," in *Proceedings of the Second Workshop on Optimizing Player Satisfaction (AIIDE 07)*, 2007.

[115] ROBERTS, D. L., STRONG, C. R., and ISBELL, C. L., "Using feature value distributions to estimate player satisfaction through an author's eyes," in *Proceedings of the AAAI 2007 Fall Symposium on Intelligent Narrative Technologies*, 2007.

[116] ROTTER, J. B., *Social learning and clinical psychology*. Prentice-Hall, 1954.

[117] ROWE, J., MCQUIGGAN, S., ROBISON, J., MARCEY, D., and LESTER, J., "StoryEval: An Empirical Evaluation Framework for Narrative Generation," in *Proceedings of the 2009 AAAI Spring Symposium on Intelligent Narrative Technologies II*, pp. 103–110, 2009.

[118] RUSSELL, S. and NORVIG, P., *Artificial Intelligence: A Modern Approach*. Prentice Hall, 3nd ed., 2009.

[119] SHARMA, M., ONTAÑÓN, S., STRONG, C., MEHTA, M., and RAM, A., "Towards Player Preference Modeling for Drama Management in Interactive Stories," in *Proceedings of the Twentieth International FLAIRS Conference (FLAIRS07)*, 2007.

[120] SPRONCK, P., SPRINKHUIZEN-KUYPER, I., and POSTMA, E., "Difficulty Scaling of Game AI," in *Proceedings of the Fifth International Conference on Intelligent Games and Simulation*, pp. 33–37, 2004.

[121] SULLIVAN, A., CHEN, S., and MATEAS, M., "Integrating drama management into an adventure game," in *Proceedings of the Fourth Annual Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE 08)*, AAAI Press, 2008.

[122] SULLIVAN, A., CHEN, S., and MATEAS, M., "From Abstraction to Reality: Integrating Drama Management into a Playable Game Experience," in *Proceedings of the 2009 AAAI Spring Symposium on Intelligent Narrative Technologies II*, 2009.

[123] SUTTON, R., PRECUP, D., and SINGH, S., "Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning," *Artificial Intelligence*, vol. 112, pp. 181–211, 1999.

[124] THALER, R. H. and SUNSTEIN, C. R., *Nude: Improving Decisions About Health, Wealth, and Happiness*. Penguin Books, 2009.

[125] THEUNE, M., KLABBERS, E., DE PIJPER, J.-R., KRAHMER, E., and ODIJK, J., "From data to speech: A general approach," *Natural Language Engineering*, vol. 7, no. 1, pp. 47–86, 2001.

[126] THUE, D. and BULITKO, V., "Modelling Goal-Directed Players in Digital Games," in *Proceedings of the Second Artificial Intelligence and Interactive Digital Entertainment Conference (AIIDE06)*, 2006.

[127] THUE, D., BULITKO, V., SPETCH, M., and WASYLISHEN, E., "Interactive storytelling: A player modeling approach," in *Proceedings of the Third Artificial Intelligence and Interactive Digital Entertainment Conference (AIIDE07)*, 2007.

[128] THURSTONE, L. L., "A law of comparative judgement," *Psychological Review*, vol. 34, pp. 273–286, 1927.

[129] THURSTONE, L. L., "Psychophysical analysis," *American Journal of Psychology*, vol. 38, pp. 368–389, 1927.

[130] VAN DEEMTER, K., KRAHMER, E., and THEUNE, M., "Real versus template-based natural language generation: A false opposition?," *Computational Linguistics*, vol. 31, no. 1, pp. 15–24, 2005.

[131] VAN DEEMTER, K. and ODIJK, J., "Context modelling and the generation of spoken discourse," *Speech Communication*, vol. 21, no. 1/2, pp. 101–121, 1997.

[132] VAN LENT, M., RIEDL, M. O., CARPENTER, P., MCALINDEN, R., and BROBST, P., "Increasing Replayability with Deliberative and Reactive Planning," in *Proceedings of the First Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE05)*, 2005.

[133] WARDRIP-FRUIN, N., MATEAS, M., DOW, S., and SALI, S., "Agency reconsidered," in *Breaking New Ground: Innovation in Games, Play, Practice and Theory, Proceedings of the Digital Games Research Association 2009 Conference (DIGRA 09)*, 2009.

[134] WEBSTER, J. M., SMITH, R. H., RHODES, A., and WHATLEY, M. A., "The effect of a favor on public and private compliance: How internalized is the norm of reciprocity?," *Basic and Applied Social Psychology*, vol. 21, no. 3, pp. 251–259, 1999.

[135] WEYHRAUCH, P., *Guiding Interactive Drama*. PhD thesis, School of Computer Science, 1997.

[136] WHITE, M. and CALDWELL, T., "EXEMPLARS: A practical, extensible framework for dynamic text generation," in *Proceedings of the Ninth International Workshop on Natural Language Generation*, pp. 266–275, 1998.

[137] WILSON, P. R., "Perceptual distortion of height as a function of ascribed academic status," *Journal of Social Psychology*, vol. 74, no. 1, pp. 97–102, 1968.

[138] YEOW, W.-L., THAM, C.-K., and WONG, W.-C., "Hard constrained semi-markov decision processes," in *Proceedings of the Twenty-First National Conference on Artificial Intelligence (AAAI06)*, (Boston, MA), 2006.

[139] YOUNG, R. M., “Notes on the Use of Plan Structures in the Creation of Interactive Plot,” in *The Working Notes of the AAAI Fall Symposium on Narrative Intelligence*, 1999.

[140] YOUNG, R. M., “An Overview of the Mimesis Architecture: Integrating Intelligent Narrative Control into an Existing Gaming Environment,” in *Working Notes of the AAAI Spring Symposium on Artificial Intelligence and Interactive Entertainment*, 2001.

[141] YOUNG, R. M., MOORE, J. D., and POLLACK, M. E., “Towards a Principled Representation of Discourse Plans,” in *Proceedings of the Sixteenth Conference of the Cognitive Science Society*, 1994.

[142] YOUNG, R. M. and RIEDL, M., “Towards an Architecture for Intelligent Control of Narrative in Interactive Virtual Worlds,” in *Proceedings of the Seventh International Conference on Intelligent User Interfaces (IUI03)*, 2003.

[143] YOUNG, R. M., RIEDL, M. O., BRANLY, M., and JHALA, A., “An Architecture for Integrating Plan-Based Behavior Generation With Interactive Game Environments,” *Journal of Game Development*, vol. 1, no. 1, pp. 54–70, 2004.

[144] ZANBAKA, C., GOOLKASIAN, P., and HODGES, L. F., “Can a virtual cat persuade you? the role of gender and realism in speaker persuasiveness,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 1153–1162, 2006.